

# **Espai de Treball per a Pràctiques**

Autor: Albert Ventura Aresté  
Directors: Juan Manuel Gimeno Illa  
Carles Mateu Pinyol

Universitat de Lleida  
Escola Politècnica Superior  
Enginyeria en Informàtica

Treball de Fi de Carrera

Juliol de 2014



# Índex

<b>Capítol 1. Introducció.....</b>	<b>7</b>
1.1.Introducció.....	8
<b>Capítol 2. Base Tecnològica del Projecte.....</b>	<b>9</b>
2.1.Gestor de Projectes Redmine.....	10
2.1.1.¿Què és Redmine? .....	10
2.1.2.Metodologia .....	10
2.1.3.Visualització general.....	11
2.1.4.Característiques Principals.....	11
2.1.5.Funcionalitat.....	14
2.1.6.Usabilitat.....	16
2.1.7.Plugins .....	17
2.1.8.Llicència .....	17
2.2.Ruby.....	18
2.2.1.Característiques .....	18
2.2.2.Semàntica.....	18
2.2.3.Sintaxis.....	19
2.2.4.Llicència.....	19
2.3.Ruby on Rails.....	20
2.3.1.Filosofia.....	20
2.3.2.Història.....	20
2.3.3.Característiques.....	20
2.4.Arquitectura Model-Vista-Controlador.....	22
2.4.1.Implantació del MVC a Ruby on Rails.....	23
<b>Capítol 3. Instal·lació i Configuració del Redmine.....</b>	<b>25</b>
3.1.Introducció.....	26
3.2.Entorn i Sistema Operatiu.....	27
3.2.1.Entorn.....	27
3.2.2.Sistema Operatiu Ubuntu 12.10.....	27
3.3.Instal·lar Redmine.....	28
3.3.1.Informació que cal saber abans d'instal·lar el Redmine.....	28
3.3.2.Instal·lació.....	28
3.4.Instal·lació i Configuració del Subversion.....	32
3.5.Crear Automàticament els Repositoris de Redmine i Autenticar els Usuaris Svn- Apache-Redmine.....	34
3.5.1.Crear Automàticament els Repositoris de Redmine.....	35
3.5.2.Autenticar els Usuaris Svn-Apache-Redmine.....	36
3.6.Enviar notifikacions de correu.....	39
<b>Capítol 4. Desenvolupament de l'aplicació: PluginUdl.....</b>	<b>41</b>
4.1.Introducció .....	42
4.2.Anàlisi de Requisits.....	43
4.2.1.Actors que hi intervenen.....	43
4.2.2.Requisits Funcionals .....	43
4.3.Models de Dades .....	45

4.3.1.Estructura de les taules de la Base de dades.....	45
4.3.2.Descripció de les taules .....	46
Taula subject_udls.....	46
Taula privilege_udls.....	46
Taula user_subject_udls.....	46
Taula project_pluginudls.....	47
Taula member_pluginudls.....	47
Taula group_user_pluginudls.....	48
4.3.3.Diagrama de Classes.....	48
4.4.Disseny Interfície Gràfica.....	50
4.4.1. Prototips d'interfícies d'usuari .....	50
4.4.2.Interfície d'Usuari.....	51
4.4.3.Thema UdL.....	57
4.5.Procés de Creació de Pràctiques mitjançant el PluginUdl.....	58
4.5.1.Descripció.....	58
4.5.2.Exemples.....	59
Creació d'una pràctica Individual.....	59
Creació d'una pràctica en Grup.....	60
4.5.3. Estàndards utilitzats per la generació de Identificadors i Noms .....	61
4.6.Implementació.....	63
4.6.1.Estructura del Plugin.....	63
4.6.2.Creació del PluginUdl .....	65
1. Crear l'estructura del PluginUdl dins al Redmine.....	65
2. Crear el Model/Controlador/Vista base de l'aplicació.....	65
4.6.3.Instal·lar PluginUdl.....	67
4.6.4.Desinstal·lar PluginUdl.....	67
4.6.5.Modificacions Internes del codi de Redmine.....	68
4.6.6.Internacionalització.....	68
4.7.Estudi del procés d'autenticació d'un Usuari.....	70
<b>Capítol 5. Conclusions i Futures línies de treball.....</b>	<b>71</b>
5.1.Conclusions tecnològiques.....	72
5.2.Conclusions Personals.....	74
5.3.Futures línies de treball.....	75
<b>Bibliografia.....</b>	<b>77</b>

## ***Índex de Figures***

Figura 1: Visualització General del Redmine.....	11
Figura 2: Vista de la creació d'un projecte mitjançant el Redmine i els seus Modules i Tràkers. .....	12
Figura 3: Vista de les característiques d'un projecte (més concretament els seus Paràmetres) .....	14
Figura 4: Esquema MVC.....	22
Figura 5: Esquema de MVC en Rails.....	24
Figura 6: Esquema de l'entorn creat entre la UdL i Redmine.....	42
Figura 7: Esquema d'interaccions entre BBDD i el PluginUdl.....	45
Figura 8: Diagrama de classes per gestionar les dades de la UdL.....	48
Figura 9: Diagrama de classes per gestionar les dades que crea el PluginUdl.....	49
Figura 10: Prototip del menú principal.....	50
Figura 11: Prototip del formulari Inserir projectes Individual.....	50
Figura 12: Prototip del formulari Inserir projectes en Grup.....	51
Figura 13: Vista de la pestanya UdL al accedir al Redmine.....	52
Figura 14: Vista de la pantalla d'inici del PluginUdl.....	52
Figura 15: Vista del formulari Inserir Projecte Individual.....	53
Figura 16: Vista del formulari Inserir Projecte en Grup.....	54
Figura 17: Vista del popup que s'utilitza per la creació dels grups de pràctiques .....	55
Figura 18: Vista del llistat, que veu el professor, de tots els projectes creats a Redmine.....	56
Figura 19: Vista del llistat, que veu l'alumne, de tots els seus projectes creats a Redmine....	57
Figura 20: Estructura de carpetes del PluginUdl.....	63



# CAPÍTOL 1. INTRODUCCIÓ

# 1.1. Introducció

Com ja sabem, les noves generacions d'estudiants que surten de la Universitat són les més formades dels últims anys, però això no significa que tinguin l'experiència i la capacitat necessària per començar a entrar en el món professional. Aquest projecte intenta acostar, una mica més, els alumnes a la realitat professional.

Es pretén que aprenguin a treballar en grup, a organitzar i administrar recursos, temps i activitats, a utilitzar gestors de versions de codi i gestors de controls d'errors, tot això mitjançant un Gestor de Projectes anomenat Redmine. La gestió de projectes és una tasca crítica en el món de les noves tecnologies, per això és molt important conèixer les característiques i problemàtiques que existeixen.

El projecte consisteix en l'elaboració d'una extensió (plugin) dintre de Redmine, per tal de crear un espai compartit via web per a les pràctiques d'una assignatura. Aquest plugin s'anomena PluginUdl i bàsicament el que fa és convertir cada grup de pràctiques d'una pràctica de l'assignatura en un projecte de Redmine, simulant per cada grup de pràctiques un entorn de projecte real. Els alumnes del grup de pràctiques seran desenvolupadors i els professors cap de projectes.

La finalitat del projecte es podria resumir en dos grans eixos que queden definits segons els usuaris que hi intervenen. L'eix més important, i que és la finalitat a la que es vol arribar, és la possibilitat de permetre als alumnes arribar a treballar amb una eina com el Redmine. L'altre eix, i no menys important, és la creació del plugin que permet als professors aplicar el Redmine per a les seves pràctiques.

Aquest treball està organitzat de la següent manera: en el segon capítol es dona una introducció als gestors de projectes, més concretament al Redmine, al seu framework de desenvolupament Ruby on Rails i al llenguatge Ruby. En el capítol tercer s'explica la instal·lació i configuració de Redmine dins d'un Sistema Operatiu Ubuntu. En el quart capítol, i el més important, s'explica tot el procés de creació del plugin i les seves característiques. En l'últim capítol s'exposaran les conclusions a les que s'han arribat.

Durant aquest treball es parla de dos termes, pràctica i projecte, que poden arribar a tenir el mateix significat però cal distingir-los per no crear confusió. Parlarem de pràctica quan el professor encara no l'hagi introduït al Redmine. Una vegada s'hagin introduït les dades al Redmine ja es parlarà de projecte.



# CAPÍTOL 2. BASE TECNOLÒGICA DEL PROJECTE

## 2.1. Gestor de Projectes Redmine

El Gestor de Projectes té com a finalitat principal la planificació, seguiment i control de les activitats i dels recursos humans i materials que intervenen en el desenvolupament d'un Sistema d'Informació. Aquest control de les activitats permet conèixer els problemes que sorgeixen i resoldre'ls de manera immediata.

El principal desafiament de la gestió de projectes és assolir tots els objectius del projecte i els objectius del temps respecte a les limitacions preconcebudes. Les limitacions principals són l'abast, temps, qualitat i pressupost. El secundari, i repte més ambiciós, és optimitzar l'assignació dels inputs necessaris i integrar-los amb els objectius predefinits.

### 2.1.1. ¿Què és Redmine?

Redmine és un gestor i planificador de projectes amb interfície web, orientat a la coordinació de tasques, comunicació de participants, i que pot especialitzar-se en projectes de desenvolupament gràcies a eines com la integració en un Repository de codi.

Està desenvolupat usant el framework Ruby on Rails i el seu disseny està significativament influenciat pel Trac, una altra eina amb característiques similars.

### 2.1.2. Metodologia

El primer que s'ha de fer quan tenim instal·lat Redmine és iniciar el treball de back-office i donar d'alta els usuaris, amb els seus diferents rols i permisos, tot i que també poden donar-se d'alta ells mateixos a través de la interfície web, perquè després l'administrador n'aprovi l'accés.

El següent pas és donar d'alta un projecte i assignar-li un cap de projectes. Una vegada completada aquesta fase podem començar a establir les etapes del mateix. Aquest pas podrà ser realitzat per tots els usuaris que tinguin els rols per fer-ho.

Per defecte la configuració de Redmine dóna quasi total accés al cap de projectes per realitzar qualsevol acció (crear projectes, editar projectes, adjuntar membres als projectes,..) per aquesta raó cal tenir en compte a qui posem com a cap de projecte. L'usuari administrador és el que té total control sobre Redmine.

Una vegada donats d'alta els projectes, els seus membres i les etapes es podran afegir les previsions de temps de cada una de les tasques i podrem obtenir el *gràfic de Gantt* per a cada etapa. Cada tasca s'assignarà a un membre de l'equip. Els membres tenen a la seva pàgina d'entrada, una llista de tasques que tenen assignades. És una única llista conjunta de les tasques de tots el projectes a on estan treballant. Segons es van fent les tasques es pot

anar marcant el temps estimat que trigaran a realitzar-la, el temps que han treballat en ella i el percentatge que creuen que tenen realitzat.

Amb aquesta informació, en l'etapa corresponent del projecte es mostra una *barra de progrés* horitzontal, on una part apareix amb color verd, indicant el % realitzat i la resta sense color, indicant el que queda pendent. Aquesta barra de progrés dóna una idea bastant aproximada del que portem fet i el que queda per fer.

Una vegada comencen les proves de software, en Redmine també es poden donar d'alta els *bugs* o errors que vagin sortint, assignant-les a desenvolupadors del projecte.

## 2.1.3. Visualització general



Figura 1: Visualització General del Redmine.

## 2.1.4. Característiques Principals

- Múltiples Projectes
- Flexibilitat en el sistema de seguiment de problemes

- Flexibilitat en el control d'accés basat en rols
- Diagrama de Gantt i Calendari
- Notícies, documents i gestió d'arxius
- Avisos mitjançant correus electrònics
- Wiki per projecte
- Fòrums per projecte
- Simplicitat en el seguiment del temps
- Integració de control de codi de Versions (SVN, CVS, Git, Mercurial, Bazar i Darcs)
- Autenticació LDAP
- Múltiples idiomes
- Varies Bases de dades
- Plugins
- Api rest

## • Projectes

Redmine gira al voltant del Projecte, igual que el Plugin de la UdL, per això cal explicar les característiques internes del projecte.

Al crear el projecte es configuraran totes les opcions i mòduls que es volen per aquell projecte.

**Projecte nou**

Nom \*

Descripció

Identificador \*

Longitud entre 1 i 100 caràcters. Only lower case letters (a-z), numbers, dashes and underscores are allowed.  
Once saved, the identifier cannot be changed.

Pàgina web

Públic ☒

Subprojecte de

Inherit members ☐

**Mòduls**

<input checked="" type="checkbox"/> Seguidor d'assumpes	<input checked="" type="checkbox"/> Seguidor de temps	<input checked="" type="checkbox"/> Notícies	<input checked="" type="checkbox"/> Documents	<input checked="" type="checkbox"/> Fixers
<input checked="" type="checkbox"/> Wiki	<input checked="" type="checkbox"/> Dipòsit	<input checked="" type="checkbox"/> Taulers	<input checked="" type="checkbox"/> Calendari	<input checked="" type="checkbox"/> Gantt

**Seguidors**

<input checked="" type="checkbox"/> Errores	<input checked="" type="checkbox"/> Tareas	<input checked="" type="checkbox"/> Soporte
---	--	---

Crea Crea i continua

*Figura 2: Vista de la creació d'un projecte mitjançant el Redmine i els seus Modules i Tràkers.*

Tot seguit s'explicaran algunes opcions del Projecte:

- Podrà ser públic amb accés a tots els usuaris de Redmine.
- Podrà ser un subprojecte d'un altre projecte.
- S'hi podran incloure tota mena de mòduls:
  - Repository: permet navegar pel Repository del projecte i veure les últimes confirmacions.
  - Wiki: és un lloc col·laboratiu que pot ser editat per tots els usuaris que formen part del projecte.
  - Forum: permet als usuaris d'un projecte comunicar-se amb els altres.
  - News: es poden penjar tota mena de notícies sobre el projecte.
  - Gantt: mostra peticions/assumptes que tenen una data d'inici i una data de venciment o estan assignades a una versió amb una data.
- Trackers:
  - Errors, Tasks, Suport: Els membres del projecte podran incloure nous assumptes/peticions que podran ser del tipus Error, Tasca o Suport.

El cap de projecte inclourà tots els membres que formaran part del projecte.

Els rols d'usuaris que existeixen per defecte són:

- Cap de Projecte
- Desenvolupador
- Informador
- NonMember
- Anonymous

L'administrador del Redmine podrà crear i editar rols. El Cap de projecte podrà editar el projecte després de crear-lo (Wiki, Repository,...).

Figura 3: Vista de les característiques d'un projecte (més concretament els seus Paràmetres)

## 2.1.5. Funcionalitat

En aquest apartat s'explicaran les funcionalitats més bàsiques i importants que té Redmine.

- **Gestor de múltiples projectes**

Redmine permet gestionar múltiples projectes des d'una sola interfície amb una finestra de navegador. La navegació és molt senzilla i es pot saltar i canviar de projecte en qualsevol moment. A més, cada projecte pot tenir una configuració totalment diferent i l'usuari pot tenir un rol diferent en cada un.

Els projectes poden ser privats, on l'administrador haurà de donar accés als membres, o públics, visibles per a tothom.

Dins de cada projecte també es podran definir subprojectes.

- **Personalització de projectes**

A Redmine cada projecte és totalment personalitzable, i podem trobar projectes molt diferents entre si segons els seus objectius.

El més important són els mòduls que hi ha en cada projecte, que poden ser activats o desactivats segons convingui: wiki, forum, notícies, peticions/assumpes, control de temps, documents, fitxers i Repository.

Exemple:

- Si un projecte està enfocat a incidències, es pot configurar per incloure només notificacions.
- Si es busca un projecte més col·laboratiu, el mòdul de wiki o el fòrum serien una bona solució.

- **Seguiment de tasques**

Una de les mecàniques més útils pel desenvolupament d'un projecte en Redmine són les peticions/assumpes i la seva visualització. Aquestes poden ser de diferents tipus (errors, tasques, suport,..) i poden assignar-se a un membre del projecte. Es pot indicar una data d'inici i final per la petició, fins i tot portar un control del temps i percentatge realitzat. També es pot assignar una prioritat i adjuntar-hi documentació. Es podran fer cerques per la majoria dels atributs que conté una petició (%realitzat, usuari assignat, prioritat,...).

Cal dir que, aquest seguiment no serà possible si l'usuari assignat no actualitza la petició cada vegada que hi treballa (temps treballat, %realitzat,...).

- **Integració en Repository de Codi**

Redmine pot integrar-se amb un Repository de codi (Subversion, Git, CVS, entre d'altres) que ha d'estar muntat en la mateixa màquina, només s'ha d'indicar el directori local. L'aplicació serveix d'interfície Web pel seguiment del desenvolupament d'un projecte.

- **Ús del calendari i diagrama de Gantt**

Redmine inclou un calendari per visualitzar totes les peticions al llarg d'un mes en concret, marcant clarament el dia d'inici i de final de cada petició.

Igualment passa amb la vista del diagrama de Gantt, que va marcant amb el percentatge completat conforme avancen els dies.

- **Notificacions**

Configurant prèviament el servidor de correu SMTP, Redmine permet enviar notificacions per correu electrònic en tots els projectes, definint abans els esdeveniments que activen aquests avisos. A més, cada usuari en la seva configuració pot escollir rebre notificacions de qualsevol esdeveniment.

- **Exportar a diferents formats**

Els informes de peticions que permeten visualitzar les diferents tasques d'un projecte, poden exportar-se en PDF o format CSV. La pàgina de la wiki pot exportar-se en HTML o TXT.

- **Altres característiques**

Redmine és una eina plena de funcionalitats i on qualsevol cosa pot configurar-se o conté opcions.

Alguna altra funcionalitat que s'hauria de destacar és la pàgina personal de cada usuari, que ofereix una vista amb tota la informació sobre els projectes en que treballa.

## **2.1.6. Usabilitat**

- **Disseny de la interfície**

El disseny de Redmine té una interfície web senzilla. Per a cada projecte existeixen una sèrie de pestanyes en la part superior que organitzen els diferents mòduls. Dins de cada mòdul es pot mostrar la informació corresponent de forma neta i ordenada. L'ús és molt simple i destaca la facilitat per configurar els projectes.

Inclou alguns temes que poden descarregar-se sense problema.

- **Fàcil ús**

L'aplicació Redmine és senzilla d'utilitzar i encara que l'àmbit principal pot ser empresarial o de desenvolupament de software, qualsevol usuari podria usar Redmine per administrar els seus propis projectes o tasques. La navegació web és molt intuïtiva i fins i tot pot recordar la d'alguns blogs.

L'estructura i la profunditat de les opcions està molt ben trobada i amb poc temps pots entendre la funcionalitat principal.

- **Problemes**

Un dels problemes més destacats que hi ha és el de les notificacions per correu, que estan adjuntades a tots els projectes. Tot i poder-se activar i desactivar sempre s'aplica a tots els projectes. Es poden configurar els esdeveniments que han d'enviar correus però no poden individualitzar-se en cada projecte.

La instal·lació i configuració de Redmine és, sense dubte, el més complicat, pels passos



(alguns delicats) que s'han de donar. Encara que hi hagi molta informació al respecte, és massa informal i no sempre dóna els resultats esperats.

## 2.1.7. Plugins

Crec que és necessari parlar sobre aquest tema, ja que la base del TFC és la creació d'un Plugin sobre el qual s'hi aplicaran totes les especificacions.

Redmine disposa d'un gran nombre de plugins que es poden trobar en el següent link:

[www.redmine.org/wiki/redmine/Plugin\\_List](http://www.redmine.org/wiki/redmine/Plugin_List)

En la llista anterior venen els autors de cada un, una petita descripció, versió compatible i a on obtenir-lo. La varietat dels plugins permet que s'adaptin a diferents necessitats. Es poden destacar alguns dedicats a generar gràfics, a establir temps per a cada tasca o a crear sales de chat.

## 2.1.8. Llicència

És de software lliure i de codi obert, disponible sota la Llicència Pública General de GNU v2 (GPL), i els seus termes poden ser consultats en el següent enllaç:

[www.gnu.org/licenses/gpl-2.0.html](http://www.gnu.org/licenses/gpl-2.0.html)

## 2.2. Ruby

Ruby és un llenguatge de programació interpretat i orientat a objectes publicat el 1995 pel programador japonès *Yukihiro Matsumoto*, també conegut com *Matz*.

*Matz* estava fascinat pel llenguatge Perl i inicialment la seva intenció era crear un Perl avançat. Però en lloc de millorar Perl, va decidir crear un llenguatge nou a partir dels seus llenguatges preferits (Perl, Lisp, Smalltalk, Eiffel i Python).

*Volia un llenguatge que fos més poderós que Perl i més orientat a objectes que Python.*

Ruby està dissenyat per la productivitat i la diversió del programador, seguint els principis d'una bona interfície d'usuari. *Matz* sosté que el disseny de sistemes necessita emfatitzar les necessitats humanes més que les de la màquina.

*Ruby és simple en aparença, però interiorment complex, com el cos humà.*

### 2.2.1. Característiques

- És un llenguatge de scripts, modern i orientat a objectes, que combina una important flexibilitat amb una alta productivitat.
- Llenguatge multi-plataforma que s'integra perfectament en gran quantitat d'arquitectures.
- Fàcilment portable: es desenvolupa majoritàriament en Linux, però funciona en diferents tipus de UNIX, Mac OS X, Windows 95/98/Me/NT/2000/XP/Vista/7, DOS,...
- Control d'excepcions per facilitar la manipulació d'errors.
- Possibilitat de redefinir els operadors.
- Àmplia llibreria estàndard.
- Monkey Patch, modificació de classes prèviament definides. Un pot reescriure, eliminar o reanomenar un mètode d'un altre.
- Mixins, Interfícies amb implementació. *Mixin* és una classe que ofereix certa funcionalitat per ser heretada per una subclasse.
- Són innecessàries les declaracions de variables, no tenen tipus i la gestió de memòria es realitza automàticament.
- Quatre nivells d'àmbit de variable: global, classe, instància i local.

### 2.2.2. Semàntica

Ruby és un llenguatge orientat a objectes. Tots els tipus de dades són un objecte, fins i tot les classes i altres com els enters i booleans. Tota funció és un mètode. Les variables sempre

són referències a objectes, no els mateixos objectes.

Ruby suporta herència. Ha estat definit com un llenguatge de programació multi-paradigma: permet la programació *Procedural*.

### 2.2.3. Sintaxis

1. La sintaxis de Ruby és similar a la de Perl i Python. La definició de classes i mètodes està definida per paraules clau.
2. La gran diferència entre Ruby i C i Perl és que les paraules claus són usades per definir blocs de codi sense claus. Els salts de línia són significatius i són interpretats com el final d'una sentència.
3. Una de les diferències entre Ruby, Python i Perl és que Ruby manté totes les variables d'instància privada dins de les classes i només s'exposen a través de mètodes d'accés (*attr\_writer*, *attr\_reader*, etc). A diferència dels mètodes *get* i *set* d'altres llenguatges com C++, c# o java, els mètodes d'accés a Ruby poden ser escrits amb una sola línia de codi. Com la crida d'aquests mètodes que no requereix l'ús de parèntesi.

### 2.2.4. Llicència

L'interpret i les biblioteques estan llicenciades de forma dual sota les llicències lliures i de codi obert GPL i llicència pública Ruby.

A partir de la versió 1.9.3 s'opta per una llicència dual sota les llicències BSD (*Berkeley Software Distribution*) de dos clàusules i llicència pública Ruby.

## 2.3. Ruby on Rails

Ruby on Rails també conegut com a RoR és un framework d'aplicacions web de codi obert escrit en llenguatge Ruby i orientat a les metodologies de web àgils.

### 2.3.1. Filosofia

RoR va se creat com a resultat de l'experiència de dos entorns de desenvolupament PHP (ràpid i brut) i Java/J2EE (lent i net), i el que es va fer va ser unir el millor d'ambdós per tal de proveir un entorn de desenvolupament web funcional, còmode i simple pel desenvolupador. Per complir amb l'esmentat en les línies anteriors Ruby on Rails es basa en els següents principis com a filosofia:

- Don't Repeat Yourself (DRY): Referit a no usar el mateix codi o les mateixes definicions en múltiples llocs de l'aplicació. RoR, per això, dona els mitjans necessaris perquè les variables i els fragments de codi que es puguin repetir ocupin un únic i accessible lloc dins de l'aplicació.
- Convention over Configuration (CoC): Significa que el programador només ha d'especificar els aspectes no convencionals de l'aplicació. Un exemple es pot trobar en la relació de les classes que representen el model de l'arquitectura MVC amb la base de dades. Per convenció, una classe del model correspondrà amb una taula de la base de dades i tindrà com a nom el singular del títol de la taula. (Per exemple per una taula que es diu *Students*, el nom de la classe model associada serà *Student*).

### 2.3.2. Història

- Ruby on Rails va ser escrit per David Heinemeier Hansson i alliberat al públic per primera vegada el juliol del 2004.
- Extret a partir del seu projecte de 37signals anomenada Basecamp.
- Actualment es troba en la versió 4.0.3 publicada el 18 de Febrer del 2014.

### 2.3.3. Característiques

- Arquitectura Model-Vista-Controlador.
- Aprofita el màxim la Meta-programació de Ruby.
- Tracta de combinar la simplicitat amb la possibilitat de desenvolupar aplicacions del món real escrivint menys codi.
- Opinionated (un framework amb opinió).

- Compte amb un potent motor de generació de codi.
- Es pot estendre la seva funcionalitat a través de plugins o gems.
- Connexió a diferents motors de base de dades.
- Permet migracions entre esquemes de la BD.
- No s'usa SQL directament en les consultes a la base de dades (però es pot).
- Unobtrusive AJAX integrat (Jquery per defecte a partir de la versió 3.1, abans prototype).
- Gran quantitat de Helpers per generar elements repetitius (formularis, camps,...)
- Compte amb una consola interactiva.
- Ús d'assets (Javascript. Css. Imatges) (a partir de la versió 3.1).
- Compte amb preprocessadors de CSS i JavaScript per defecte (CoffeScript i Sass).
- Conté varies tasques *rake* predefinides per operacions comunes com migracions, test,..
- Utilitza les rutes de manera fàcil i dinàmica.
- Suport integrat a Internacionalització (i18n) i Localització.

## 2.4. Arquitectura Model-Vista-Controlador

L'arquitectura Model-Vista-Controlador (MVC) és un patró d'arquitectura de les aplicacions de software. La seva principal característica és la d'aïllar la interfície d'usuari d'una aplicació de la seva lògica interna, amb l'objectiu de facilitar la modificació individual de qualsevol d'aquestes dues parts sense que l'altre es vegi afectada.

El nom d'aquesta arquitectura ve donat pels seus tres components:

- El **model**, es refereix a tota la informació que manipula l'aplicació i gestiona tots els accessos a aquesta informació, tant consultes com actualitzacions, implementant també tots els privilegis d'accés.
- La **vista**, són tots aquells components de l'aplicació que són visibles per l'usuari i amb els que pot interactuar.
- El **controlador** és aquell que enllaça els dos anteriors, modificant la informació que es troba en el model d'acord amb la informació que rep des de la vista.

Com es pot veure, cadascú d'aquests components manipula elements de naturalesa diferent, per tant la seva funcionalitat bàsica es pot aïllar fàcilment. Tot i així, si han de formar part d'una mateixa aplicació, necessitaran compartir la mateixa informació.

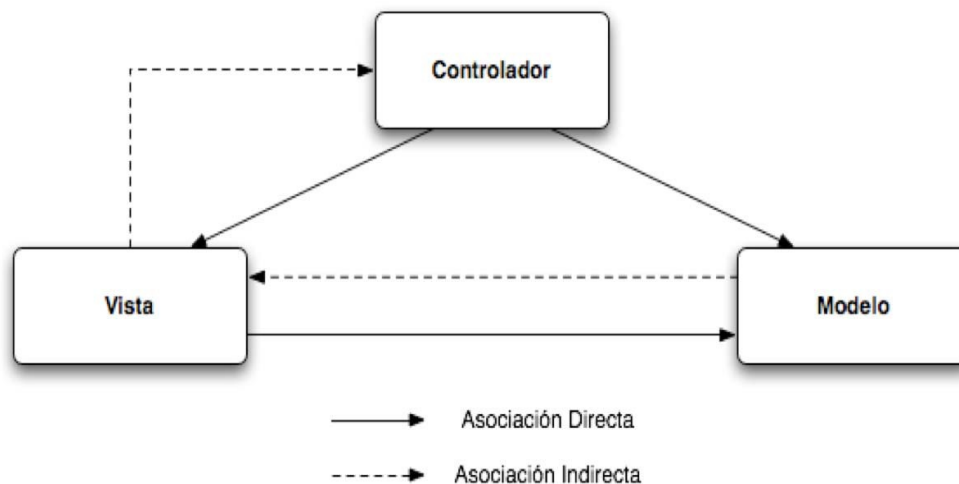


Figura 4: Esquema MVC

## 2.4.1. Implantació del MVC a Ruby on Rails

- **Model**

El model proporciona una interfície i es vincula amb les taules de la BD. Conté el codi del Programa Ruby que manipula els registres de la BD.

A Ruby on Rails, les classes del Model són desplegades per la llibreria ActiveRecord. L'únic que ha de fer el programador es heretar de la classe ActiveRecord::Base, i el programa esbrinarà automàticament quina taula utilitzar i quines columnes té.

El model representa:

- Les taules de la BD
- Migracions (expressen canvis a l'esquema DB)

- **Vista**

La vista és la lògica de visualització i defineix plantilles per la representació de dades. A les aplicacions Web, la vista consisteix en una quantitat mínima de codi en HTML.

Existeixen en l'actualitat moltes maneres de gestionar les vistes. La metodologia que s'utilitza a Rails per defecte és usar Ruby "empotrat", que són bàsicament fragments de codi HTML amb codi Ruby. La llibreria Rails utilitzada per definir les plantilles de presentació és ActionView.

- **Controlador**

És el que s'encarrega de la interacció entre la interfície de BD (ActiveRecord) i el motor de la presentació (ActionView), és a dir, responen a les accions dels usuaris mitjançant la vista que a la vegada manipula dades de les classes del Model i mostra els resultats usant les Vistes. La llibreria que utilitza és ActionController i proporciona instal·lacions per la manipulació i l'organització de les dades de la BD i/o de formulari web d'entrada.

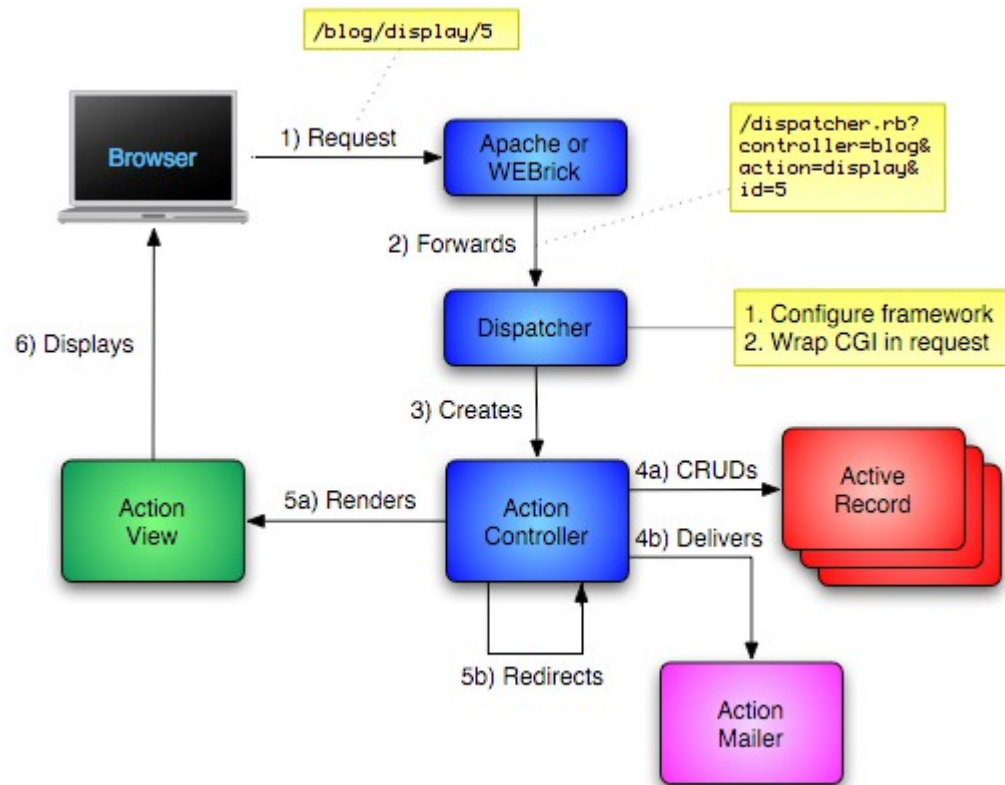


Figura 5: Esquema de MVC en Rails.



# CAPÍTOL 3. INSTAL·LACIÓ I CONFIGURACIÓ DEL REDMINE

## 3.1. Introducció

La instal·lació del Redmine consta de diverses fases:

- Entorn utilitzat pel Projecte.
- La instal·lació del Sistema Operatiu
- Instal·lació de diversos softwares específics per la utilització de Redmine (Ruby, Framework Rails, Base de Dades, Servidor de pàgines Web,..)
- Configuració del Redmine pel correcte funcionament amb el plugin.

## 3.2. Entorn i Sistema Operatiu

En aquest apartat s'indicarà l'entorn i el software utilitzat per poder desenvolupar tot el projecte.

### 3.2.1. Entorn

Per tal de treballar en un entorn còmode i exportable s'ha creat una màquina virtual mitjançant el programa Vmware Workstation. Tot seguit s'indiquen totes les eines i programes utilitzats pel projecte:

Software	Versió
VMWare Workstation	9.0.2 build-1031769
Ubuntu	12.10
Redmine	2.3
MySQL	5.5.35
Ruby	1.9.3
Rails	3.2.13
RubyMine	6.3.2

### 3.2.2. Sistema Operatiu Ubuntu 12.10

La instal·lació del Sistema Operatiu Ubuntu s'ha realitzat com ja s'ha comentat, en una màquina virtual, s'ha triat un sistema Linux perquè Windows ha donat alguns problemes alhora de treballar amb Redmine. La màquina virtual s'ha creat amb les següents configuracions:

- 3GB Memòria
- 4 Processadors
- 150GB Disc Dur

Tot i tenir aquestes configuracions, durant el transcurs del projecte s'han anat canviant les configuracions (2GB, 2 Processadors).

## 3.3. Instal·lar Redmine

La versió instal·lada per aquest projecte és la versió 2.3 (en l'actualitat Redmine va per la 2.5.2). Per internet es poden trobar diferents manuals per instal·lar Redmine per les diferents versions d'Ubuntu.

### 3.3.1. Informació que cal saber abans d'instal·lar el Redmine

Base de dades que suporta Redmine:

- **Mysql 5.0 o superior**

És important instal·lar els enllaços de C per a Ruby perquè millora el rendiment. Es pot instal·lar mitjançant la següent ordre:

```
gem install mysql2
```

- **PostgreSQL 8.2 o superior**

En aquesta instal·lació s'ha d'assegurar que el "datestyle" de la base de dades s'estableix en la norma ISO. Es pot configurar mitjançant:

```
ALTER DATABASE "redmine_db" SET datestyle = "ISO, MDY";
```

A les versions de PostgreSQL 8.4.0 i 8.4.1 sorgeixen alguns errors que afecten el comportament del Redmine.

- **Microsoft SQL Server 2008 o Superior (des de Redmine 2.3.0)**

- **SQLite 3 (no es pot utilitzar en mode multi-usuari a producció)**

### 3.3.2. Instal·lació

- **Actualitzar Ubuntu a la última versió**

```
sudo apt-get update
```

- **Instal·lar el servidor Apache i el mòdul Passenger.**

Passenger serveix pel desplegament d'aplicacions implementades en Ruby, fins i tot aquelles aplicacions construïdes amb framework Ruby on Rails (Redmine). És un conducte entre l'Apache i l'aplicació Ruby.

```
sudo apt-get install apache2 libapache2-mod-passenger
```

- **Instal·lar els programes bàsics que necessita Redmine per la instal·lació.**

```
sudo apt-get install mysql-server mysql-client  
sudo apt-get install redmine-mysql
```

- **Instal·lar els paquets bàsics i les dependències.**

```
sudo apt-get install ruby rubygems libruby ruby-dev
```

Rubygems és un Repository públic on estan allotjades pràcticament totes les gemmes públiques que hi ha a Ruby. També és un gestor de les gemmes que facilitarà la instal·lació d'aquestes.

Les gemmes son plugins, llibreries, Extensions i/o codis afegits als projectes de Ruby on Rails, que ens permeten noves funcionalitats, noves funcions predefinides,...

- **Descarregar l'ultima versió del Redmine**

```
cd /usr/local/share/  
sudo svn co http://svn.redmine.org/redmine/branches/2.3-stable redmine-2.3  
sudo ln -s /usr/local/share/redmine-2.3 /usr/local/share/redmine  
sudo chown -R root:root /usr/local/share/redmine-2.3
```

- **Llibreries necessàries per a MYSQL**

```
sudo apt-get install libmysqlclient-dev libmagickcore-dev libmagickwand-dev
```

- **Instal·lar Bundler**

Una vegada tenim instal·lat el RubyGems, podem instal·lar la gemma Bundler que ens permetrà a partir d'un fitxer (Gemfile), poder definir totes les gemmes que utilitzarem en un projecte.

```
sudo gem install bundler
```

Una vegada tenim el bundler instal·lat a l'Ubuntu, l'haurem d'utilitzar per instal·lar totes les gemmes del projecte Ruby on Rails. En cas de començar un nou projecte s'haurien d'afegir a l'arxiu Gemfile totes les gemmes que es necessitarien per desenvolupar el projecte. En aquest cas Redmine ja ha inclòs al document Gemfile totes les necessàries per treballar. L'únic que es farà serà no instal·lar o excloure les que no s'utilitzaran.

```
cd /usr/local/share/redmine/  
bundle install --without development test postgresql sqlite (Instal·lar totes les gemmes del  
Gemfile excepte les excoles).
```

Una de les gemmes més importants que instal·larà serà el mysql2 que és la que ens permetrà connectar MySQL amb Redmine.

### • Crear Base de Dades

Accedir al Mysql:

```
mysql -u root -p
```

Crear Base de dades Redmine:

```
create user 'redmine' identified by 'redmine';  
set password for 'redmine'@'localhost' = password('my_password');  
grant all on *.* to 'redmine'@'localhost';  
create database redmine;  
quit;
```

### • Configuració General

Entrar a la carpeta /usr/local/share/redmine/config i crear un arxiu amb el nom de database.yml. Aquest arxiu el crearem a partir del /usr/local/share/redmine/config/database.yml.example

```
sudo cp database.yml.example database.yml
```

Editar database.yml i modificar les dades segons el que s'hagi posat en la creació de la BD.

*production:*

```
adapter: mysql2 (nota: Per Rails < 3.1 usa mysql enlloc de mysql2)  
database: redmine  
host: localhost  
username: redmine  
password: my_password
```

Generate a session store secret:

```
rake generate_secret_token
```

Generate the database structure:

```
RAILS_ENV=production rake db:migrate
```

Generate default configuration data:

```
RAILS_ENV=production rake redmine:load_default_data
```

- **Encriptació de la Base de dades (Opcional)**

Si es vol xifrar la base de dades de Redmine s'haurà de fer el següent:

```
sudo cp configuration.yml.example configuration.yml
```

Modificar el configuration.yml posant un \* al database\_cipher\_key:

```
database_cipher_key: *
```

Un cop modificat l'arxiu executem la següent comanda:

```
rake db:encrypt RAILS_ENV=production
```

- **Configuració Apache**

Configuració de l'arxiu default que es troba a la carpeta:

Modificar el VirtualHost del Redmine

```
gedit /etc/apache2/sites-available/default
```

Afegir el següent codi:

```
<VirtualHost *:8080>  
  ServerName redmine.example.com  
  DocumentRoot /usr/local/share/redmine/public  
  <Directory /usr/local/share/redmine/public>  
    AllowOverride all  
    Options -MultiViews  
  </Directory>  
</VirtualHost>
```

- **Comprovar el funcionament de Redmine**

Entrar a la carpeta on hi ha instal·lat el Redmine, en el nostre cas (/usr/local/share/redmine/) i executar la següent comanda:

```
sudo ruby script/rails server webrick -e production
```

## 3.4. Instal·lació i Configuració del Subversion

Redmine es pot integrar, a més del Subversion, a una gran varietat de controls de versions (CVS, Git, Mercurial, Bazar i Darcs).

### - Instal·lar subversion:

```
sudo apt-get install subversion libapache2-svn
```

El següent pas seria configurar l'accés dels usuaris al subversion i això es fa mitjançant:

```
sudo htpasswd /etc/subversion/passwd albert  
sudo htpasswd -c /etc/subversion/passwd albert
```

El símbol “-c” s'utilitza per sobrescriure el password d'un usuari ja creat.

**- Crear la carpeta arrel del Repository de Redmine**, en aquest cas es crea a la ruta */var/svn*, i donar els permisos pertinents per tal de que sigui accessible per Apache.

```
mkdir /var/svn  
chown -R www-data:www-data /var/svn
```

Cada vegada que es vulgui crear un Repository d'un projecte Redmine s'haurà de crear una nova carpeta a l'arrel svn i després executar la següent comanda per convertir-lo en un Repository.

```
mkdir /var/svn/myproject  
svnadmin create /var/svn/myproject
```

Existeix un mètode per crear de forma automàtica els Repositories al crear el Projecte (veure apartat 3.5).

Per aconseguir accedir als seus Repositories des d'una direcció web, s'haurà d'afegir el següent codi a la configuració Apache ubicada generalment a */etc/apache2/site-available*.

```
<Location /svn>  
DAV svn  
SVNPath /var/svn  
AuthType Basic  
AuthName "Redmine Repository"  
AuthUserFile /etc/subversion/passwd
```



*Require valid-user*  
*</Location>*

- **Reiniciar l'Apache:** *sudo service apache2 restart*

## 3.5. Crear Automàticament els Repositoris de Redmine i Autenticar els Usuaris Svn-Apache-Redmine

En aquest apartat s'explicarà com crear automàticament els Repositoris dels projectes de Redmine i com configurar el Redmine per tal que només puguin accedir als Repositoris els usuaris que tenen els permisos per fer-ho.

Hi ha dos maneres de fer-ho:

- **Usant `apache/mod_dav_svn/mod_perl`**

És l'opció més senzilla i ràpida de configurar, però té l'inconvenient de que en el cas que hi hagi Repositoris grans o molts Repositoris aquesta opció perd rendiment. Això és degut a que utilitza l'eina *mod\_perl*, molt útil però molt deficient quan treballa amb dades grans.

- **Usant mòduls `pam` i `nss`**

Aquesta opció té una configuració molt més complexa, però el seu rendiment és molt millor. Utilitza la configuració dels mòduls *PAM* per permetre que els membres del projecte tinguin permisos de lectura/escriptura, *svnserve* perquè els usuaris anònims puguin llegir projectes públics i *apache/mod\_dav* per permetre la navegació Redmine. Això només funciona en Unix amb els mòduls PAM i una base de dades MySQL.

En aquest cas s'ha configurat l'**opció 1**, tot i que en futures línies de treball seria bo configurar la segona opció.

Tot seguit s'expliquen els passos que s'han dut a terme per configurar l'opció 1:

Apache/mod\_dav\_svn/mod\_perl

### 3.5.1. Crear Automàticament els Repositories de Redmine

Des de la versió 0.5.0, Redmine és capaç de manipular la creació d'un Repository Subversion. Això es fa mitjançant el reposman.rb que és una seqüència de comandes (script) que es troben a la carpeta extra/svn. A partir de la versió de Redmine 0.6.0 reposman.rb pot registrar el nou Repository i establir el propietari.

- **Els projectes es recuperen de Redmine utilitzant un servei web SOAP.** Aquest servei web està desactivat per defecte en Redmine. Per activar-lo només s'ha d'anar a:

*Administració → Configuració → Repository → (Habilitar WS per la gestió del Repository)*

- **Per tal de crear els Repositories dels projectes que encara no en tenen només s'ha d'executar la següent comanda al terminal.**

```
ruby /usr/local/share/redmine/extra/svn/reposman.rb --redmine localhost --svn-dir /var/svn
--owner www-data --url http://localhost/svn/ --key=nRmc4N8c6TgaND25k1GY --verbose
```

La key és única per cada Redmine i la podrem trobar:

*Administració → Configuració → Repository → (Key)*

- **El problema de l'apartat anterior és que per tal de crear els Repositories dels projectes haurem d'executar manualment la comanda.** Per automatitzar aquest procés s'utilitzarà l'eina cron, que ens permet programar processos per a que s'executin en una hora i data determinada.

*El crontab és una eina programada de Linux.*

```
gedit /etc/crontab
```

incloent la següent línia:

```
***** root ruby /usr/local/share/redmine/extra/svn/reposman.rb --redmine localhost --svn-
dir /var/svn --owner www-data --url http://localhost/svn/ --key=nRmc4N8c6TgaND25k1GY
>> /var/log/reposman.log
```

Els (\*) signifiquen els moments en què vols executar aquesta línia. En aquest cas s'executarà sempre. Tot seguit s'explica un exemple per entendre-ho millor:

**1 10 15 \* \* 1s ---> En el minut 1, a las 10 del matí, els dies 15, de qualsevol mes, i qualsevol dia de la setmana s'executarà 1s.**

## 3.5.2. Autenticar els Usuaris Svn-Apache-Redmine

En aquest apartat s'explicarà la configuració que s'ha dut a terme per tal de que els usuaris que formen part d'un projecte puguin entrar al Repository.

Hi ha dos maneres de treballar amb Repositories:

- Repositories locals. La ruta d'accés al Repository serà:

<file:///svn/redmine> //on Redmine és un Repository

- Repositories no locals, on ens trobem amb alguns problemes d'accés. La ruta d'accés al Repository serà:

<http://localhost/svn/redmine> //on Redmine és un Repository

El nostre servidor està pensat per treballar de forma no local i per tant, tal i com s'ha dit anteriorment, existixen alguns problemes d'accés. Aquests venen donats pel fet que els usuaris registrats al Redmine no tenen privilegis per entrar al Apache-Subversion. La solució serà que els usuaris de Redmine estiguin al Apache i d'aquesta manera cada vegada que un usuari vol accedir al Subversion es comprovarà la seva autenticació al Redmine. Tot seguit s'explicaran els passos per configurar aquesta opció.

Cal dir que als Repositories dels projectes que són públics hi podrà accedir tothom.

- **Instal·lar les llibreries necessàries**

```
sudo apt-get install libapache2-svn libapache-dbi-perl libapache2-mod-perl2 libdbd-  
mysql-perl  
sudo apt-get install libdigest-sha1-perl
```

- **Comprovar que tots els mòduls estiguin “enable”**

```
sudo a2enmod dav  
sudo a2enmod dav_svn # if you want to use svn  
sudo a2enmod dav_fs # if you want to use git  
sudo a2enmod perl
```

– **Configuració del Apache per Repositories de Subversion**

Copiar l'arxiu Redmine.pm d'on tenim instal·lat el Redmine  
 “/usr/local/share/redmine/extra/svn” a la carpeta “/usr/lib/perl5/Apache/”.

```
cp /usr/local/share/redmine/extra/svn/Redmine.pm /usr/lib/perl5/Apache/
```

– **Canviar el codi de configuració del Apache, “/etc/apache2/sites-available/default”:**

```
<VirtualHost *:80>
  ServerName localhost
  DocumentRoot /usr/local/share/redmine/public
  PerlLoadModule Apache::Redmine
  <Location /svn>
    DAV svn
    SVNParentPath "/var/svn"
    Order deny,allow
    Deny from all
    Satisfy any

    #LimitXMLRequestBody 0
    #SVNPathAuthz off

    PerlAccessHandler Apache::Authn::Redmine::access_handler
    PerlAuthenHandler Apache::Authn::Redmine::authen_handler

    AuthType Basic
    AuthName "Redmine SVN Repository"
    AuthUserFile /etc/subversion/passwd
    #read-only access
    <Limit GET PROPFIND OPTIONS REPORT>
      Require valid-user
      #Allow from 127.0.0.1
      #Satisfy any
    </Limit>
    # write access
    <LimitExcept GET PROPFIND OPTIONS REPORT>
      Require valid-user
    </LimitExcept>

    ## Mysql-Settings
    RedmineDSN "DBI:mysql:database=redmine;host=localhost"
    RedmineDbUser "redmine"
```

```
    RedmineDbPass ""  
</Location>  
</VirtualHost>
```

Per tal de poder controlar d'una manera més rigorosa l'accés als Repositories s'ha limitat l'accés de lectura als usuaris que tenen accés al Redmine i en cas de ser un projecte privat als usuaris que pertanyen al projecte.

```
<Limit GET PROPFIND OPTIONS REPORT>  
    Require valid-user  
    #Allow from 127.0.0.1  
    #Satisfy any  
</Limit>
```

## 3.6. Enviar notificacions de correu

Cal dir que l'opció d'enviar un correu de notificació quan un projecte es crea no està desenvolupada al Redmine. Hi ha un assumpte pendent de fer des del 2008 a la pàgina d'assumptes de Redmine.

Tot i això hi ha moltes altres opcions de notificacions per correu com per exemple:

- Issue added
- Issue update
  - Issue note added
  - Issue status updated
  - Issue priority updated
- News added
- Document added
- File added
- Message added
- Wiki page added
- Wiki page updated

Tot seguit s'explicarà la configuració que s'hauria d'utilitzar per poder configurar les notificacions. No s'ha implementat al Redmine pel fet de no tenir un gestor de correus per fer proves. La següent configuració està feta amb un correu personal però només s'hauria de canviar i configurar-ho amb el Gestor de Correus de la UdL.

1. Agafar l'arxiu `/usr/local/share/redmine/config/configuration.yml.exemple` i posar-lo a la mateixa carpeta amb diferent nom `/usr/local/share/redmine/config/configuration.yml`
2. Editar l'arxiu `configuration.yml`
  1. Primer de tot comentar tota la informació que hi ha a l'arxiu. Es diu comentar i no eliminar, perquè seria bo tenir una configuració d'exemple en el cas de que es vulguin fer modificacions.
  2. Afegir la següent informació.

```
production:
  delivery_method: :smtp
  smtp_settings:
    tls: true
    enable_starttls_auto: true
    address: "smtp.gmail.com"
    port: '587'
```

```
domain: "smtp.gmail.com"  
authentication: :plain  
user_name: quildu123@gmail.com  
password: "Contrasenya del Correu"
```

3. Editar l'arxiu `/usr/local/share/redmine/config/environment.rb` i canviar la següent línia:

```
config.action_mailer.perform_deliveries = false  
per  
config.action_mailer.perform_deliveries = true
```

4. Habilitar un parell de coses al Redmine:

Administration --> Settings --> Incoming emails --> *Enable WS for incoming emails*

Administration --> Settings --> Email Notifications



# CAPÍTOL 4. DESENVOLUPAMENT DE L'APLICACIÓ: PLUGINUDL

## 4.1. Introducció

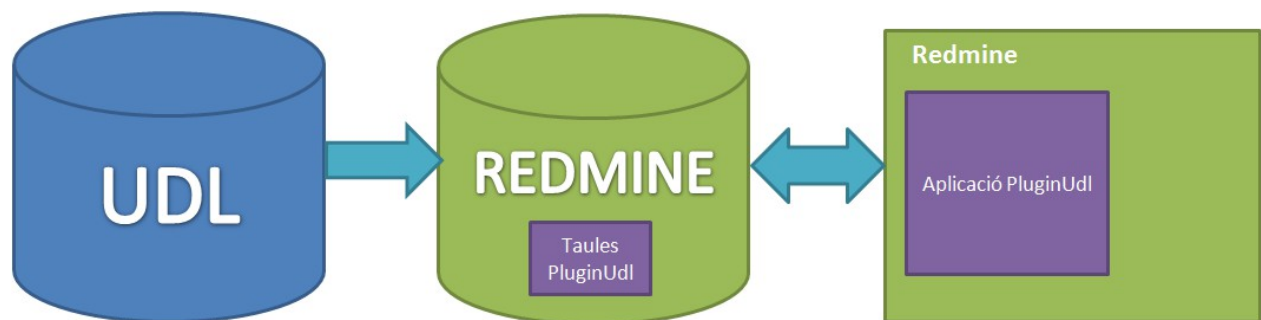
Redmine com s'ha dit anteriorment és un gestor de projectes molt utilitzat per les empreses tecnològiques, que l'utilitzen per administrar projectes complexos, controlant-ne en cada moment la seva execució.

La intenció del projecte, al marge de que els alumnes coneguin i treballin amb un gestor de projectes, és crear una eina adaptada al Redmine que permeti simular un entorn real a totes les pràctiques de les assignatures de la UdL. Cada pràctica d'un alumne o grup d'alumnes d'una assignatura es convertirà en un projecte únic dins del Redmine.

L'eina utilitzada per definir i implementar les especificacions és un Plugin de Redmine (anomenat *PluginUdl*).

Es pretén crear un entorn entre la UdL i el Redmine per tal d'agilitzar la creació de les pràctiques d'una assignatura com a projectes de Redmine. No es vol crear un projecte específic i diferent als projectes de Redmine sinó que es vol aprofitar la gran capacitat de treball que té, per utilitzar-lo a les pràctiques.

El PluginUdl és la part més important de l'entorn ja que s'encarrega de la creació dels projectes, però també cal destacar la importància que tenen les taules de la BD ja que faciliten la creació de l'entorn de treball entre la UdL i el Redmine. En aquest capítol s'explicaran aquests dos punts.



*Figura 6: Esquema de l'entorn creat entre la UdL i Redmine*

Perquè el PluginUdl pugui treballar de forma àgil i ràpida s'han de controlar i guardar molt bé les dades que s'utilitzaran. Existeixen dos tipus de taules, les creades per gestionar i emmagatzemar les dades dels projectes del PluginUdl i les encarregades de guardar totes les dades de la UdL.

Una vegada es tenen totes les dades guardades, el professor podrà crear les pràctiques de les assignatures amb les configuracions pertinents mitjançant el PluginUdl.

## 4.2. Anàlisi de Requisits

Per poder crear un entorn per a totes les pràctiques s'ha de fer un estudi i un anàlisi del funcionament intern de la UdL, més concretament de les pràctiques.

### 4.2.1. Actors que hi intervenen

Abans de començar a pensar en possibles requisits, resulta convenient definir quins seran els usuaris del sistema. Els usuaris es podran dividir segons el benefici que en treuen.

Els professors seran els beneficiaris directes del PluginUdl, ja que els permetrà crear les pràctiques de les assignatures com a projectes dins de Redmine d'una manera molt ràpida i senzilla.

Per altra banda, l'alumne és el beneficiari final, ja que treballaran amb una eina molt potent com és el Redmine, i aprendran a treballar en grup, a gestionar els seus propis projectes i a utilitzar eines indispensables per gestionar-los, com per exemple l'eina de control de versions (Subversion).

### 4.2.2. Requisits Funcionals

Els requisits funcionals fan referència a totes aquelles característiques de comportament que ha d'incloure el PluginUdl i tot seguit se'n mostren algunes d'elles.

#### Metodologies de treball a les pràctiques

S'han definit dos tipus de sistemes o metodologies de treball dins de les pràctiques:

- Individual: els alumnes fan les pràctiques de forma individual.
- En Grup: els alumnes treballen en grups de pràctiques.

#### Tipus i Rols d'usuaris en els projectes de Redmine

Dins de cada pràctica hi intervenen dos tipus d'usuaris, el professor, que és el que revisa i controla el procés de la pràctica i l'alumne o alumnes que són els que van desenvolupant el codi de les pràctiques i fent consultes al professor si tenen dubtes. D'aquí en resulten dos rols d'usuaris:

- Cap de Projecte (Professor)
- Desenvolupador (Alumne)

Cal remarcar que els Alumnes no tindran mai el rol de cap de projectes en un projecte, ja que això els permetria tenir el mateix rol que un professor de forma permanent.

#### Creació d'un espai compartit

En una pràctica sempre hi ha coses compartides o comunes per a tots els alumnes, ja sigui l'enunciat, documentació extra,... Així que s'haurà de crear un lloc comú per a cada pràctica, i la solució ha estat crear un Projecte principal per a cada pràctica d'on penjaran tots els

projectes dels alumnes, i a partir d'ara aquests s'anomenaran subprojectes. El projecte principal estarà format per tots els usuaris de l'assignatura, els professors com a cap de projectes i els alumnes com a desenvolupadors (*Veure exemple de creació d'una pràctica*).

### **Gestió del codi**

Aquest plugin està pensat per assignatures que treballen amb informació digital, per aquesta raó es necessita un Repository per emmagatzemar-la. De forma automàtica, per cada projecte es crea el seu respectiu Repository, i això es fa mitjançant la configuració del Redmine (Veure Capítol 3.5). Només tenen accés als Repositories els usuaris que formen part del projecte.

Per tal de controlar les diferents versions de la informació del Repository s'utilitza el Control de Versions *Subversion*. Aquesta opció es podria deshabilitar si l'administrador de Redmine treu el Repository com a característica per defecte de Redmine.

### **Gestió dels Grups de pràctiques d'una assignatura**

Sovint els grups de pràctiques es mantenen durant tota una assignatura i per aquesta raó s'ha creat una taula que els guarda per poder-los utilitzar en la creació de les altres pràctiques de l'assignatura.

### **Privilegis dintre el PluginUdl**

Els únics usuaris que podran crear projectes al Redmine són els professors. Per aquest motiu s'ha creat una taula *privilege\_udls* que ens defineix si un usuari és o no professor. Professor (privilegi = 1), Alumne (privilegi = 0).

### **Cap de Projectes**

En una pràctica hi pot haver un o més professors. Tots poden fer el paper de cap de projectes, i és per aquest motiu que l'aplicació permetrà seleccionar més d'un professor.

### **Característiques de la Pràctica**

Les característiques (Wikis, Repository) que tindran els projectes creats seran les que té per defecte Redmine. Així un dels punts a millorar en futures ampliacions és poder definir les característiques que vol un professor per cada projecte.

## 4.3. Models de Dades

En aquest apartat s'explicarà tota l'estructura i la relació del model de dades del PluginUdl.

### 4.3.1. Estructura de les taules de la Base de dades

Totes les taules són creades pel PluginUdl. S'han creat models per a cada taula i així al instal·lar el Plugin les taules es creen automàticament.

Tot seguit es mostren els diferents tipus de taules que existeixen segons el seu paper dins del Plugin. Per tal de diferenciar-ho, s'ha creat per cada tipus, un estàndard pel nom.

- Taules creades per gestionar i emmagatzemar les dades dels projectes de la UdL creats al Redmine. Aquestes taules s'utilitzen bàsicament per tenir un control dels projectes creats. No es vol tenir una rèplica dels projectes de Redmine ja que en cap cas es creen les taules Wiki, Repository,... La única taula important serà la group\_PluginUdls, que permetrà extreure els grups de projectes d'una assignatura en concret.
  - member\_pluginudls, project\_pluginudls i group\_PluginUdls
- Taules encarregades de guardar totes les dades de la UdL necessàries perquè tant el PluginUdl com el Redmine les puguin utilitzar.  
 Inicialment es va voler crear un programa per tal de traspasar totes les dades de la UdL al Redmine, tal i com es fa al Sakai, però degut a la dificultat de poder accedir a la BD de la UdL, el que s'ha fet és crear un conjunt d'usuaris i assignatures directament a la BD de Redmine. Tot i això, s'han analitzat i utilitzat els camps reals que utilitza la UdL.  
 No es crea una taula users\_udl per guardar els usuaris, ja que després d'analitzar els camps necessaris es va arribar a la conclusió que la taula users de Redmine era la solució.
  - subject\_udls, user\_subject\_udls i privilege\_udls

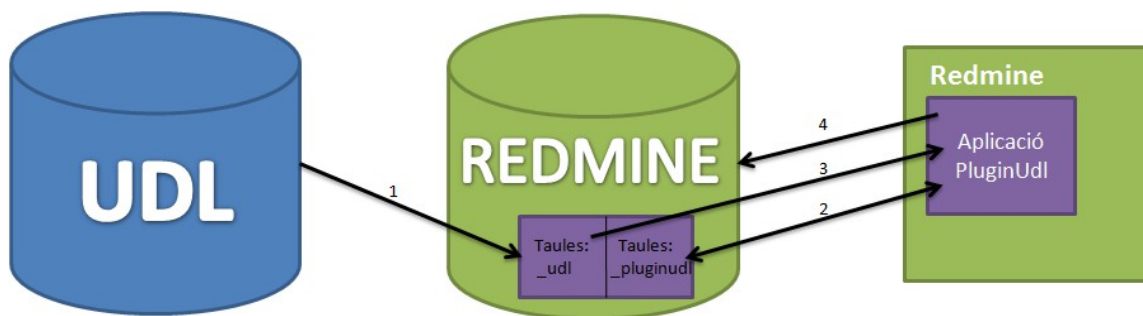


Figura 7: Esquema d'interaccions entre BBDD i el PluginUdl

## 4.3.2. Descripció de les taules

Els noms de les taules i els seus camps segueixen els estàndards de Redmine.

### Taula `subject_udls`

Emmagatzema totes les assignatures de la UdL. Aquestes assignatures s'insereixen manualment mitjançant el phpMyAdmin. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la clau primària.
- **name:** és el nom de l'assignatura.
- **description:** és la descripció de l'assignatura.
- **code:** és el codi únic que té l'assignatura a la UdL.
- **center:** és el centre al que pertany de la UdL (EPS, Lletres, Medicina,...).
- **academic\_year:** és una variable que indica l'any acadèmic de l'assignatura per tal de poder filtrar les assignatures correctes. (Ex: 2013-2014).
- **enabled:** ens diu si una assignatura està habilitada o no. Pot passar que tingui un any acadèmic correcte però no es vulgui utilitzar per crear Projectes.
- **created\_on:** és la data en que s'insereix a la base de dades de Redmine.

### Taula `privilege_udls`

Defineix els privilegis que té un usuari de Redmine. Els privilegis en aquest cas poden ser alumnes o professors. Aquests privilegis s'insereixen manualment mitjançant el phpMyAdmin. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la clau primària.
- **user\_id:** és l'identificador del usuari.
- **privilege:** és 1 si l'usuari és professor i 0 si és alumne.

### Taula `user_subject_udls`

Com que la relació entre la taula `subjects_udls` i `users` és de *many to many* s'ha creat una taula que les relaciona (Veure Figura 8). Aquestes relacions s'insereixen manualment mitjançant el PhpMyAdmin. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la clau primària.
- **user\_id:** és l'identificador de l'usuari.
- **subject\_udl\_id:** és l'identificador de l'assignatura.

## Taula project\_pluginudls

La taula project\_pluginudls ha de tenir els mateixos elements que la taula projects de Redmine. És la taula que conté una còpia de tots els projectes que es creen mitjançant el PluginUdl. Els camps *lft*, *rgt*, *update\_on* són elements de la taula projects que no estan a la taula project\_pluginudls, ja que s'introdueixen de forma automàtica al crear el projecte. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la clau primària.
- **name:** és el nom del projecte, que serà el mateix nom, tant a la taula projects de Redmine com a la taula project\_pluginudls.
- **identifier:** cada projecte té un identificador únic i, per això, aquest identificador haurà de ser únic. De moment, encara no s'ha parlat de quines paraules s'agafaran per compondre-lo. (Veure apartat 4.5.3)
- **description:** és una breu descripció del projecte.
- **public:** indica si al projecte hi tindrà accés tothom o només els usuaris de Redmine. Serà 1 si és públic i 0 si és privat.
- **parent\_id:** és l'identificador del project\_pluginudls principal.
- **status:** indica l'estat en que està un projecte (habilitat o deshabilitat). En aquest moment no s'utilitza per res a la taula project\_pluginudls, ja que els projectes del PluginUdl només són una còpia (no han d'estar habilitats ni deshabilitats). Serà 1 si està habilitat i 0 si està deshabilitat. Aquest camp es podria actualitzar en el moment de deshabilitar el projecte de Redmine.
- **copy\_status:** ens informarà si el projecte dins de project\_pluginudls ha estat copiat al Redmine. El valor serà 1 si s'ha copiat i 0 sinó s'ha copiat.
- **created\_on:** és la data que en què s'insereix a la taula un registre. No té cap connexió amb el camp *created\_on* de la taula projects, ja que aquest es genera automàticament.
- **subject\_udl\_id:** és el id de l'assignatura relacionada amb aquest projecte.

## Taula member\_pluginudls

La Taula member\_pluginudls conté els membres relacionats a un projecte de la taula project\_pluginudls. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la seva clau primària.
- **user\_id:** és l'identificador de l'usuari.
- **project\_pluginudl\_id:** és l'identificador del project\_pluginudls al qual pertany.
- **role\_id:** és el rol que té aquest membre dins del projecte. Agafa els valors de la taula *roles* de Redmine. Serà 3 si el membre és *cap de projecte* i 4 si és *desenvolupador*.
- **copy\_status:** ens informarà si el membre dins de member\_pluginudls ha estat

copiat al Redmine. El valor serà 1 si s'ha copiat i 0 sinó s'ha copiat.

## Taula group\_user\_pluginudls

Aquesta taula guarda els grups d'alumnes que es generen al crear els projectes. Els seus camps són:

- **id:** és un camp que es genera sol al crear la taula, és la clau primària.
- **name:** és el nom del project\_pluginudls al qual pertany.
- **group\_id:** és el número de grup al qual pertany dins del project\_pluginudls.
- **project\_pluginudl\_id:** és l'identificador del project\_pluginudls al qual pertany.
- **user\_id:** és l'identificador del usuari.

### 4.3.3. Diagrama de Classes

El diagrama de classes descriu l'estructura d'un sistema mostrant les classes, atributs i les relacions entre ells. A continuació, s'exposarà una breu descripció del PluginUdl.

S'han creat dos diagrames de classes que defineixen els dos tipus de taules que existeixen segons els seu paper en el Plugin. Es podria haver creat un únic diagrama però, d'aquesta manera es pot entendre millor el funcionament del PluginUdl.

El següent diagrama representa les classes i relacions necessàries per guardar i treballar amb les dades de la UdL.

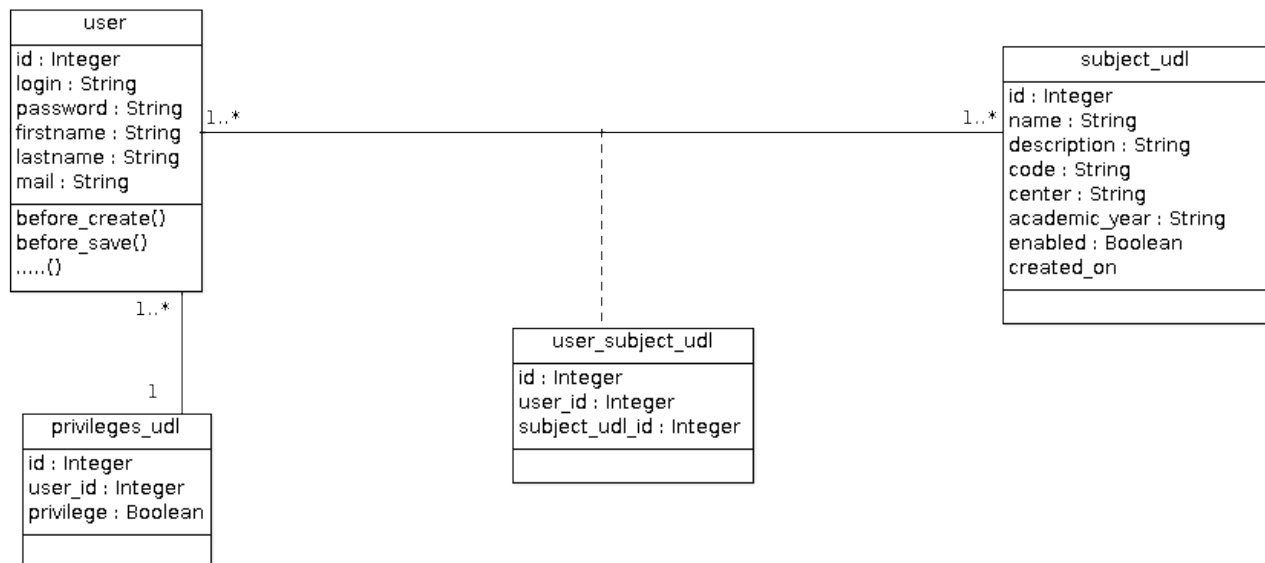


Figura 8: Diagrama de classes per gestionar les dades de la UdL



Aquesta part està representada per:

- *user (Redmine)*: gestiona els usuaris de la UdL.
- *subject\_udl*: assignatures de la UdL.
- *user\_subject\_udl*: la relació entre usuari i assignatura és de n a n. Aquest tipus de relació requereix una classe unió que conté les seves claus principals com a claus externes.
- *privilege\_udl*: privilegis (professor o alumne).

El següent diagrama representa les classes i relacions necessàries per guardar i treballar amb les dades dels projectes creats pel PluginUdl.

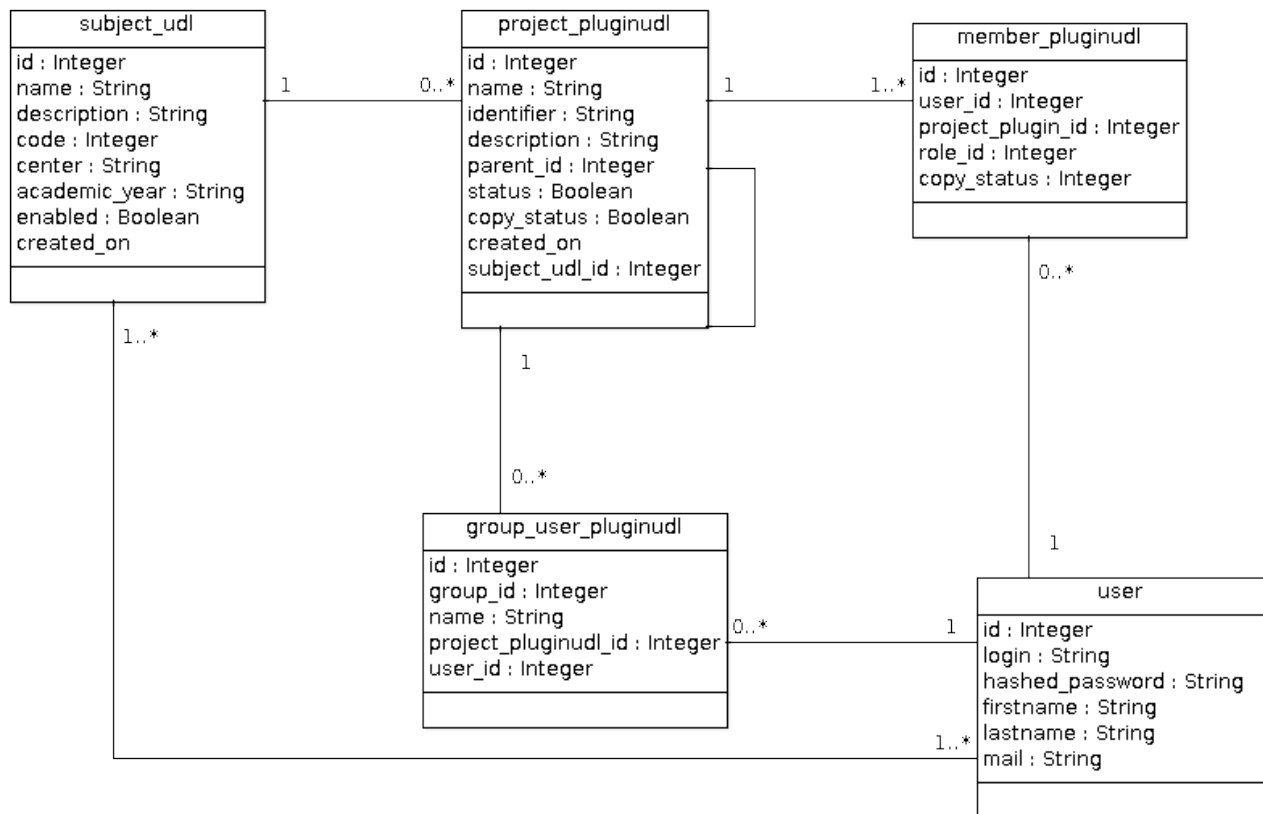


Figura 9: Diagrama de classes per gestionar les dades que crea el PluginUdl

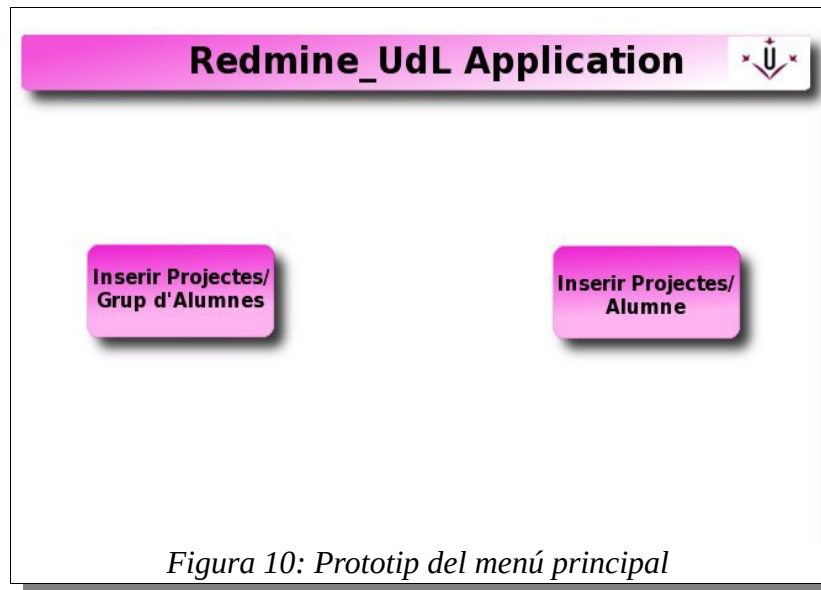
Aquesta part està representada per:

- *project\_pluginudl*: projectes creats pel PluginUdl.
- *member\_pluginudl*: membres que pertanyen als projectes. Té un atribut role\_id que a Redmine està relacionat amb la classe Roles i en aquest cas només es posen els valors però no es relacionen.
- *group\_user\_pluginudl*: grups creats a un projecte. Si un projecte és individual no tindrà grup.
- *user (Redmine)*: usuaris de la UdL que són els membres dels projectes
- *subject\_udl*: cada projecte està assignat a una assignatura.

## 4.4. Disseny Interfície Gràfica

### 4.4.1. Prototips d'interfícies d'usuari

#### - Menú Principal



*Figura 10: Prototip del menú principal*

Navegabilitat: En aquesta finestra es mostrarà el menú principal del Plugin on es podrà escollir el tipus de projectes que es vol crear.

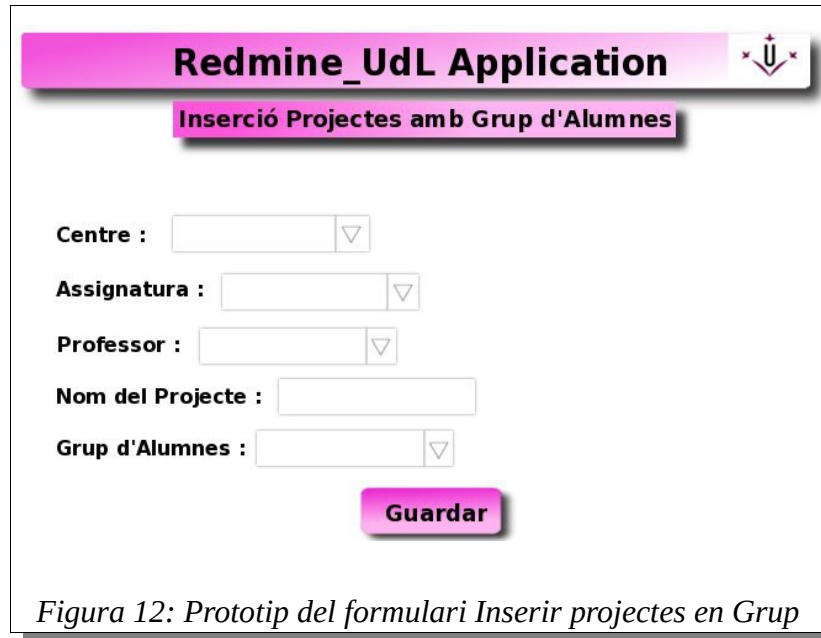
#### - Inserir Projecte Individual

El prototip del formulari per inserir projectes individuals. A la part superior hi ha una barra de títol magenta amb el text "Redmine\_UdL Application" i un icona d'usuari a la dreta. Just a sota, hi ha una barra magenta amb el text "Inserció de Projecte per Alumne". El formulari conté quatre camps d'entrada: "Centre :" amb un menú desplegable, "Assignatura :" amb un menú desplegable, "Professor :" amb un menú desplegable, i "Nom del Projecte :" amb un camp de text. A la part inferior dreta hi ha un botó magenta amb el text "Guardar".

*Figura 11: Prototip del formulari Inserir projectes Individual*

Navegabilitat: Aquesta finestra constarà d'un formulari on el professor introduirà les dades per crear els nou Projectes/Alumnes.

### - Inserir Projectes en Grup



The image shows a software prototype window titled "Redmine\_UdL Application" with a sub-header "Inserció Projectes amb Grup d'Alumnes". It contains a form with five fields: "Centre", "Assignatura", "Professor", "Nom del Projecte", and "Grup d'Alumnes", each with a dropdown arrow. A pink "Guardar" button is at the bottom right. The caption below reads "Figura 12: Prototip del formulari Inserir projectes en Grup".

Navegabilitat: Aquesta finestra constarà d'un formulari on el professor introduirà les dades per crear els nou Projectes/Grup d'alumnes.

## 4.4.2. Interfície d'Usuari

Basant-se amb els prototips d'interfície creats anteriorment s'han creat les següents interfícies:

### Breu Descripció

La interfície està formada per una pestanya UdL, que es mostra a la part superior del Redmine, sota del menú principal. La pestanya conté el menú principal del PluginUdl. En aquest menú els professor podran triar quin tipus de pràctiques volen crear (Individual o en grup).

Individual: conté tots els atributs necessaris per crear el projecte (assignatura, professors, nom, identificador, descripció del projecte i la característica públic ).

Grup: conté tots els atributs necessaris per crear el projecte (assignatura, professors, nom, identificador, apartat de creació de grups, descripció del projecte i la característica públic ).

## Interfície de pestanya UdL

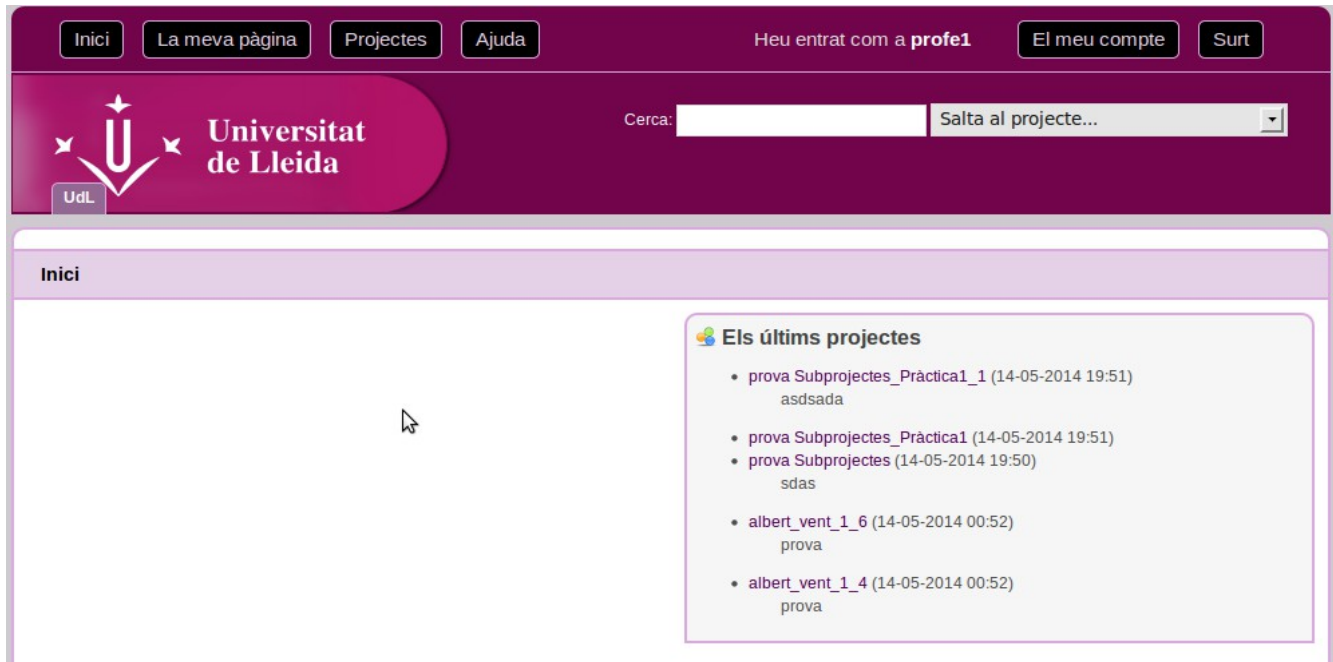


Figura 13: Vista de la pestanya UdL al accedir al Redmine

## Interfície pantalla d'inici

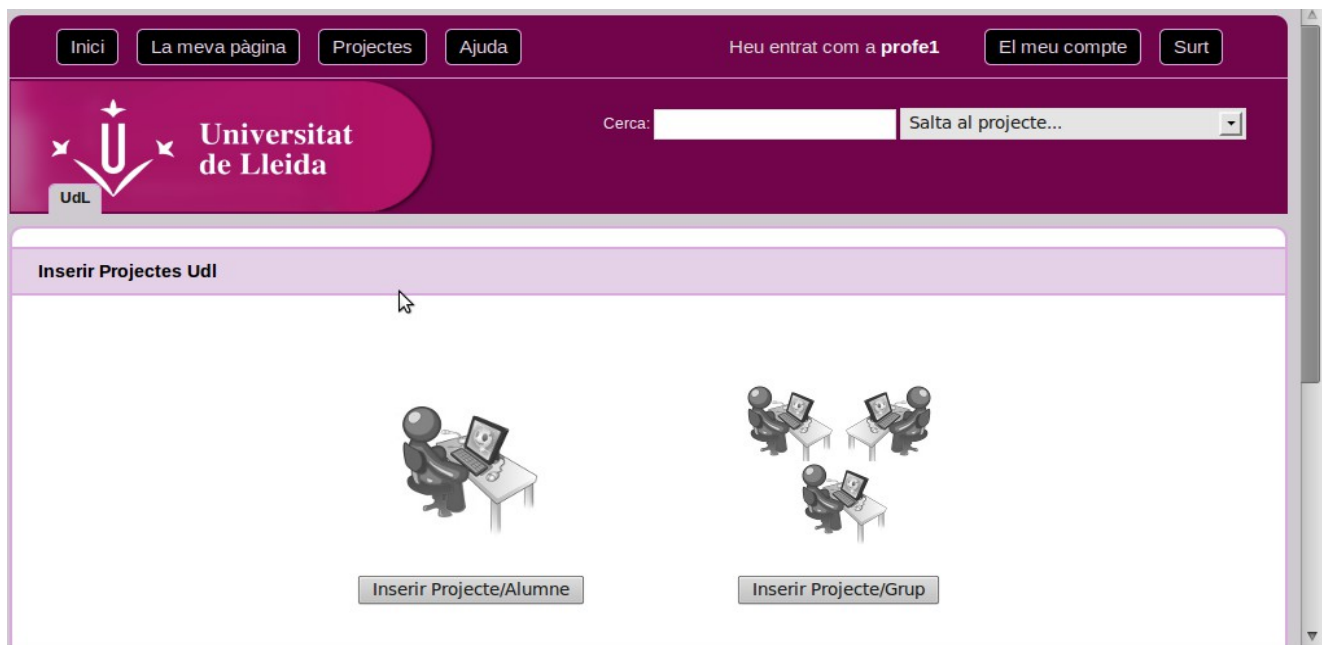


Figura 14: Vista de la pantalla d'inici del PluginUdl

## Interfície formulari Inserir Projecte Individual

**Nou Projecte Individual**

**Assignatura \***  

ESOFT

**Professors \***  

Carles Mateu  
Roberto García  
Josep Ribó

*Per seleccionar més d'un professor utilitzar la tecla control o arrastar el ratolí*

**Nom del Projecte \***  

Pràctica 1




























*30 caràcters com a màxim.*

**Identificador \***  

practica-1

*30 caràcters com a màxim.*

**Descripció**  

**B** **I** **U**  **C** **H1** **H2** **H3**                            

## Interfície formulari Inserir Projecte en Grup

## Nou Projecte Grup

**Assignatura \***

ESOFT

**Professors \***

Carles Mateu  
Roberto García  
Josep Ribó

*Per seleccionar més d'un professor utilitzar la tecla control o arrastar el ratolí*

**Nom del Projecte \***

Pràctica 2

*30 caràcters com a màxim.*

**Identificador \***

practica-2

*30 caràcters com a màxim.*

**Grups d'Alumnes \***









Afegir Grup

**Grup:**

- ☒ albert ventura
- ☒ gemma pons

Eliminar Grup

**Descripció**

B I U  C H1 H2 H3     ppt   

Públic ☐

Desa Enrere

*Figura 16: Vista del formulari Inserir Projecte en Grup*

## Vista de la creació dels Grups de pràctiques dins del formulari Inserir Projecte en Grup

The image shows a web form for creating practice groups. The form is titled "Inserir Projecte en Grup". It contains several sections:

- Assignatura \***: A dropdown menu with "ESOFT" selected.
- Professors \***: A list box showing "Carles Mateu", "Roberto Garcia", and "Josep Ribó". Below it, a note says "Per seleccionar més d'un professor utilitzar la tecla control o arrastar el ratolí".
- Nom del Projecte \***: A text input field with a note "30 caràcters com a màxim".
- Identificador \***: A text input field with a note "30 caràcters com a màxim".
- Grups d'Alumnes \***: A section with a "Afegir Grup" button and a "Grup:" label. Below it, two students are listed with checked checkboxes: "albert ventura" and "gemma pons". There is also an "Eliminar Grup" button.
- Descripció**: A rich text editor with a toolbar containing icons for bold, italic, underline, strikethrough, color, background color, bulleted list, numbered list, link, unlink, and a "pre" button.

On the right side, there is a modal window titled "Llistat d'Alumnes" (List of Students). It contains a list of students with checkboxes: "albert ventura", "gemma pons", "Joan Ventura", and "Anna Llopis". At the bottom of the modal are "Desa" (Save) and "Cancel·lar" (Cancel) buttons.

Figura 17: Vista del popup que s'utilitza per la creació dels grups de pràctiques

## Interfície pantalla llistat de Projectes al Redmine

Projectes	
★ <b>Prac 1_1234EE</b>	Prova individual
★ Prac 1_1234EE_albert	Prova individual
★ Prac 1_1234EE_gemma	Prova individual
★ Prac 1_1234EE_Joan	Prova individual
★ Prac 1_1234EE_Anna	Prova individual
★ <b>Prac 1_1234FF</b>	Prova
★ Prac 1_1234FF_group0	Prova
★ Prac 1_1234FF_group1	Prova
★ <b>Practica 1_1234FF</b>	dasda
★ Practica 1_1234FF_albert	dasda
★ Practica 1_1234FF_Anna	dasda
★ <b>Pràctic 4_1234EE</b>	dasda
★ Pràctic 4_1234EE_group0	dasda
★ Pràctic 4_1234EE_group1	dasda
★ Pràctic 4_1234EE_group2	dasda
★ <b>Pràctica 1_1234EE</b>	Prova 1
★ Pràctica 1_1234EE_group0	Prova 1
★ Pràctica 1_1234EE_group1	Prova 1
★ Pràctica 1_1234EE_group2	Prova 1
★ <b>Pràctica 4_0898FGH</b>	Prova de senzill
★ Pràctica 4_0898FGH_albert	Prova de senzill
★ Pràctica 4_0898FGH_gemma	Prova de senzill
★ <b>Pràctica 4_1234EE</b>	Prova
★ Pràctica 4_1234EE_group0	Prova
★ Pràctica 4_1234EE_group1	Prova
★ Pràctica 4_1234EE_group2	Prova

Figura 18: Vista del llistat, que veu el professor, de tots els projectes creats a Redmine



Projects	
★ <b>Prac 1_1234EE</b>	Prova individual
★ Prac 1_1234EE_albert	Prova individual
★ <b>Prac 1_1234FF</b>	Prova
★ Prac 1_1234FF_group0	Prova
★ <b>Practica 1_1234FF</b>	dasda
★ Practica 1_1234FF_albert	dasda
★ <b>Pràctic 4_1234EE</b>	dasda
★ Pràctic 4_1234EE_group0	dasda
★ <b>Pràctica 1_1234EE</b>	Prova 1
★ Pràctica 1_1234EE_group2	Prova 1
★ <b>Pràctica 2_1234EE</b>	prova
★ Pràctica 2_1234EE_group0	prova
★ <b>Pràctica 4_0898FGH</b>	Prova de senzill
★ Pràctica 4_0898FGH_albert	Prova de senzill
★ <b>Pràctica 4_1234EE</b>	Prova
★ Pràctica 4_1234EE_group1	Prova

Figura 19: Vista del llistat, que veu l'alumne, de tots els seus projectes creats a Redmine

### 4.4.3. Thema UdL

Tal i com es pot veure al disseny de la interfície s'ha creat un *theme* per la UdL. Aquest s'ha fet a partir d'una plantilla de Redmine i s'ha modificat per agafar un disseny semblant a la web de la UdL. Un vegada creat el tema només s'ha d'instal·lar al Redmine, i per instal·lar-lo només s'ha de posar a la carpeta:

```
/usr/local/share/redmine-2.3/public/themes
```

Per seleccionar el tema dins de Redmine, l'usuari administrador haurà d'anar a:

*Administration - Settings - Display*

## 4.5. Procés de Creació de Pràctiques mitjançant el PluginUdl

### 4.5.1. Descripció

Un vegada analitzades les característiques que tindrà la pràctica, els professors accediran al Redmine amb el seu nom d'usuari i contrasenya. A dalt a l'esquerra hi podran veure la pestanya UdL que forma part del PluginUdl. La pestanya conté el menú principal. En aquest menú el professor tria el tipus de pràctica que vol realitzar (en grup o individual). El formulari i procés de creació d'ambdós tipus no varia gaire.

- Formulari Individual: el professor accedeix al formulari de creació de projectes individuals i selecciona l'assignatura on vol crear la pràctica (només podrà crear projectes sobre assignatures on ell és professor). Una vegada seleccionada l'assignatura, s'haurà de seleccionar quin o quins professors seran Caps de Projectes (pot passar que una assignatura tingui més d'un professor i només un sigui el cap de projectes). Finalment només s'hauran d'introduir totes les característiques que tindrà el projecte (Nom del Projecte, Descripció i Públic).
- Formulari en Grup: el procés de creació és molt similar i la única cosa que canvia és la creació dels grups de pràctiques de l'assignatura. Quan s'ha decidit l'assignatura i els caps de projectes, s'introdueixen tots els grups de pràctiques que es crearan a partir dels usuaris de l'assignatura. Finalment, igual que a la creació individual, s'introduiran totes les característiques que tindrà el projecte (Nom del Projecte, Descripció i Públic).

Una vegada emplenades totes les característiques de la pràctica, el professor ja podrà guardar les dades i crear els projectes mitjançant el botó *guardar* que hi ha al final del formulari. És llavors quan el PluginUdl comença un conjunt de procediments interns:

#### 1. Guardar una còpia dels projectes creats a les taules del PluginUdl:

- Guardar el Projecte Principal a la taula *project\_pluginudls* amb l'atribut *parent\_id* = null. Aquest projecte principal estarà format per tots els usuaris i professors de l'assignatura i s'utilitzarà com a espai compartit.
- Guardar els membres del Projecte Principal a la taula *member\_pluginudls*.
- Guardar els Subprojectes a la taula *project\_pluginudls* amb l'atribut *parent\_id* = *id\_projecte\_principal*.
- Guardar els membres de tots els Subprojectes a la taula *member\_pluginudls*.

- Els rols dels usuaris dins de cada projecte s'inclouran a la mateixa taula *member\_pluginudls* a l'atribut *role\_id*.
- En cas de ser un projecte en grup es guardaran totes les dades dels grups de pràctiques a la taula *group\_user\_pluginudls*. Aquests grups estaran relacionats amb els seus respectius subprojectes.

## 2. Crear tots els projectes creats pel PluginUdl a Redmine.

- A partir del *id\_projecte\_principal* anterior es copiarà, juntament amb tots els seus subprojectes, a la taula *projects* de Redmine creant d'aquesta manera els projectes.
- Copiar els membres corresponents als projectes creats anteriorment a la taula *members* de Redmine. També es crearan els rols de cada usuari dins del projecte a la taula *member\_roles*.

Després de crear aquests projectes, els alumnes podran accedir al Redmine, veure'ls i començar a desenvolupar el codi per pujar-los al Repository personal del projecte. El professor per la seva banda podrà controlar en qualsevol moment quina és l'evolució i resoldre qualsevol dubte de forma personalitzada.

Comentar que el PluginUdl només es podrà visualitzar quan els professors s'hagin loguejat al Redmine.

## 4.5.2. Exemples

### Creació d'una pràctica Individual

El professor vol fer la Pràctica 1 de l'assignatura d'Edalg que té com a codi 7878AAA. Les assignatures i els seus usuaris hauran d'estar prèviament copiats i relacionats a la base de dades de Redmine.

1. El professor accedeix al seu compte de Redmine mitjançant el seu Usuari i Contrasenya.
2. A l'entrar al Redmine accedirà a la pestanya UdL.
3. Escollirà quina metodologia vol per la seva pràctica, si individual o en grup. En aquest cas escollirà, individual.
4. Una vegada dins de l'apartat de pràctiques individuals seleccionarà l'assignatura que vol fer la pràctica (Edalg). Com que no ha agafat l'opció en grup no necessitarà crear cap grup, l'únic que haurà d'omplir són les característiques que haurà de tenir la pràctica (Públic, Descripció, Nom de la Pràctica) i els professors o professor que farà de cap de projecte.
5. Guardarà la pràctica mitjançant el botó guardar i automàticament el Plugin crearà els diferents Projectes a Redmine:

- Projecte principal que estarà format per tots els usuaris de l'assignatura amb un rol de desenvolupador i pel professor o professors com a cap de projectes.

nom -> Pràctica 1\_7878AAA

identificador -> practica-1\_7878aaa

Subprojecte

nom -> Pràctica 1\_7878AAA\_alumne1

identificador -> practica-1\_7878aaa\_alumne1

professor -> cap de projectes

l'alumne1 -> desenvolupador.

Subprojecte

nom -> Pràctica 1\_7878AAA\_alumne2

identificador -> practica-1\_7878aaa\_alumne2

professor -> cap de projectes

l'alumne2 -> desenvolupador.

Subprojecte

nom -> Pràctica 1\_7878AAA\_alumne3

identificador -> practica-1\_7878aaa\_alumne3

professor -> cap de projectes

l'alumne3 -> desenvolupador.

6. Els projectes es crearan amb les característiques que té definides per defecte al Redmine. Aquestes només les podrà modificar l'administrador de Redmine.

## Creació d'una pràctica en Grup

El professor vol fer la Pràctica 2 de l'assignatura Xarxes II que té com a codi 7878BBB. En aquest cas l'assignatura té 6 alumnes.

1. El professor accedeix al seu compte de Redmine mitjançant el seu Usuari i Contrasenya.
2. A l'entrar al Redmine accedirà a la pestanya UdL.
3. Escollirà quina metodologia vol per la seva pràctica, si individual o en grup. En aquest cas escollirà, en grup.
4. Un cop dins de l'apartat de pràctiques en grup seleccionarà l'assignatura que vol fer la pràctica (Xarxes II). Seleccionarà els professors i introduirà el nom (Pràctica 2).
5. Es crearan els grups de pràctiques, que seran {(alumne1, alumne2), (alumne3, alumne4), (alumne5, alumne6)}.
6. Omplir les característiques que haurà de tenir la pràctica (Públic i Descripció).
7. Guardarà la pràctica mitjançant el botó guardar i automàticament el Plugin crearà els diferents Projectes a Redmine:

- Projecte principal que estarà format per tots els usuaris de l'assignatura amb un rol de desenvolupador i pel professor o professors com a cap de projectes.

nom -> Pràctica 1\_7878BBB

identificador -> practica-1\_7878bbb

Subprojecte

nom -> Pràctica 1\_7878BBB\_grup0

identificador -> practica-1\_7878bbb\_0

professor -> cap de projectes

l'alumne1 i alumne2 -> desenvolupadors

Subprojecte

nom -> Pràctica 1\_7878BBB\_grup1

identificador -> practica-1\_7878bbb\_1

professor -> cap de projectes

l'alumne3 i alumne4 -> desenvolupadors

Subprojecte

nom -> Pràctica 1\_7878BBB\_grup2

identificador -> practica-1\_7878bbb\_2

professor -> cap de projectes

l'alumne5, alumne6 -> desenvolupadors

8. Els projectes es crearan amb les característiques que té definides per defecte al Redmine. Aquestes només les podrà modificar l'administrador de Redmine.

### 4.5.3. Estàndards utilitzats per la generació de Identificadors i Noms

Tots els projectes creats dins de Redmine necessiten un identificador únic. Poden tenir el mateix nom, però sempre hauran de tenir diferent identificador.

Redmine crea l'identificador en base al nom del projecte amb uns petits canvis, per aquesta raó és quasi impossible que hi hagi dos projectes amb noms iguals. Tot i que l'usuari pot modificar manualment l'identificador Redmine el crea automàticament mitjançant javascript (`project_identifier.js`).

El Plugin utilitza el mateix mètode que Redmine però incloent petits canvis.

Tot i que, Redmine permet posar el mateix nom a diferents projectes, per crear el nom dels projectes s'han utilitzat estàndards que permeten diferenciar-los. Això es fa perquè a partir d'un nom es generen diferents projectes (projecte principal i tots els subprojectes).

Tot seguit s'explicaran els estàndards utilitzats per crear els noms i identificadors de tots els tipus de projectes. La variable *identifier* és la que es genera a partir del nom del projecte, igual que al Redmine i *name* és el nom que posa l'usuari al projecte.

- Projecte Principal:

Identificador: *identifier* + "\_" + *code\_subject* (tots els caràcters estaran amb minúscula.)

Nom del Projecte: *name* + "\_" + *code\_subject*

- Subprojecte/Grup:

Identificador: *identifier* + "\_" + *code\_subject* + "\_" + *numGroup* (tots els caràcters estaran amb minúscula.)

Nom del Projecte: *name* + "\_" + *code\_subject* + "\_group" + *numGroup*

- Subprojecte/Individual:

Identificador: *identifier* + "\_" + *code\_subject* + "\_" + *login\_usuari* (tots els caràcters estaran amb minúscula.)

Nom del Projecte: *name* + "\_" + *code\_subject* + "\_" + *login\_usuari*

## 4.6. Implementació

### 4.6.1. Estructura del Plugin

L'estructura de directoris del plugin té tot l'aspecte d'una aplicació Rails normal i bàsicament ho és, però està dissenyada per ser muntada sobre una altre aplicació, en aquest cas és Redmine, i per això hi ha algunes diferències. El plugin utilitza el helper script de Rails per crear l'estructura de directoris completa.

▼ app	4 items folder	Wed 25 Jun 2014 07:30:36 PM CEST
▶ controllers	1 item folder	Tue 08 Jul 2014 12:37:13 AM CEST
▶ helpers	1 item folder	Mon 30 Jun 2014 08:50:54 PM CEST
▶ models	6 items folder	Tue 08 Jul 2014 12:44:08 AM CEST
▶ views	1 item folder	Wed 25 Jun 2014 07:34:20 PM CEST
▼ assets	3 items folder	Wed 25 Jun 2014 07:30:36 PM CEST
▶ images	2 items folder	Wed 25 Jun 2014 07:37:25 PM CEST
▶ javascripts	5 items folder	Wed 09 Jul 2014 08:38:35 PM CEST
▶ stylesheets	1 item folder	Tue 08 Jul 2014 07:38:01 PM CEST
▼ config	2 items folder	Tue 08 Jul 2014 12:36:04 AM CEST
▶ locales	3 items folder	Tue 22 Jul 2014 11:48:18 AM CEST
routes.rb	787 bytes Ruby script	Tue 08 Jul 2014 12:36:04 AM CEST
▼ db	1 item folder	Wed 25 Jun 2014 07:30:36 PM CEST
▶ migrate	6 items folder	Wed 25 Jun 2014 07:33:23 PM CEST
▼ lib	1 item folder	Wed 25 Jun 2014 07:30:36 PM CEST
▶ tasks	0 items folder	Wed 25 Jun 2014 07:30:36 PM CEST
▶ test	5 items folder	Wed 25 Jun 2014 07:30:36 PM CEST
init.rb	607 bytes Ruby script	Wed 25 Jun 2014 08:02:54 PM CEST
README.rdoc	36 bytes plain text document	Wed 25 Jun 2014 07:30:36 PM CEST

Figura 20: Estructura de carpetes del PluginUdl

### ***/app***

En aquest directori s'organitzen els components del plugin. Té subdirectoris que conté la vista (vista i helpers), els controladors (controllers) i la lògica de negoci (models).

### ***/app/controller***

Un controller gestiona un requeriment web de l'usuari. Pel plugin s'ha creat un únic controlador que gestiona totes les peticions.

### ***app/helpers***

Aquesta carpeta conté la classe helper que assisteix a la classe model, view i controller. Això ajuda a mantenir el codi reduït i enfocat. En el plugin només s'ha creat una funcionalitat, utilitzada per crear el número de grup en cada moment.

### ***app/models***

Conté totes les classes que modelen les dades emmagatzemades a la base de dades, és a dir en el plugin hi ha una classe per cada taula creada.

### ***app/view***

Conté totes les plantilles per completar amb les dades del plugin i mostrar-ho al browser del usuari.

### ***/assets***

Aquesta carpeta conté tots els arxius estàtics o que no canvien del plugin com, per exemple, javascripts (javascripts/), imatges (images/) i styles (stylesheets/). Aquesta carpeta queda allotjada directament a la carpeta public de Redmine a la ruta (public/plugin\_assets/PluginUdl) i permet així, poder accedir a aquests documents des de tot el Redmine.

### ***/config***

Conté la poca quantitat de codi de configuració que necessitarà el plugin, el direccionament dels web request entrants (routes.rb) i els arxius pel funcionament de la internacionalització (locals/).

### ***/db***

En aquest directori es mantenen els scripts per la gestió de la base de dades relacional.



## 4.6.2. Creació del PluginUdl

### 1. Crear l'estructura del PluginUdl dins al Redmine

Primer de tot s'ha de crear el plugin de Redmine, i els passos a seguir per crear el plugin són:

- Establir la variable d'entorn `RAILS_ENV` amb el valor correcte, sinó es treballarà amb l'entorn per defecte, que és `development`.

```
$ export RAILS_ENV=production
```

- Crear el nou plugin usant el generador de plugins de Redmine (`generate`), això ho farem dins de l'aplicació Redmine (`/usr/local/share/redmine`):

```
ruby script/rails generate redmine_plugin PluginUdl
```

El plugin el podrem trobar a la carpeta `plugins` de Redmine (`/usr/local/share/redmine/plugins`) amb el nom `PluginUdl`. El plugin es crea amb tota l'estructura d'un projecte RoR per començar a desenvolupar el codi. Es pot ajustar la informació del plugin (nom, autor, descripció i versió) a l'arxiu `PluginUdl/init.rb`.

Un cop creat el plugin s'han de crear els models, les vistes i controladors corresponents.

### 2. Crear el Model/Controlador/Vista base de l'aplicació

- **Crear el model `project_pluginudl`**

```
ruby script/rails generate redmine_plugin_model PluginUdl project_pluginudl name:string  
identifier:string description:string public:integer parent_id:integer status:integer  
copy_status:integer created_on:datetime subject_udl_id:integer
```

```
rake redmine:plugins:migrate
```

- **Crear el model `member_pluginudl`**

```
ruby script/rails generate redmine_plugin_model PluginUdl member_pluginudl user_id:integer  
project_pluginudl_id:integer role_id:integer copy_status:integer
```

```
rake redmine:plugins:migrate
```

- **Crear el model group\_user\_pluginudl**

```
ruby script/rails generate redmine_plugin_model PluginUdl group_user_pluginudl  
group_id:integer name:string project_pluginudl_id:integer user_id:integer
```

```
rake redmine:plugins:migrate
```

- **Crear el model subject\_udl**

```
ruby script/rails generate redmine_plugin_model PluginUdl subject_udl name:string  
description:string code:string center:string academic_year:string enabled:integer  
created_on:datetime
```

```
rake redmine:plugins:migrate
```

- **Crear el model privilege\_udl**

```
ruby script/rails generate redmine_plugin_model PluginUdl privilege_udl user_id:integer  
privilege:integer
```

```
rake redmine:plugins:migrate
```

- **Crear el model user\_subject\_udl**

```
ruby script/rails generate redmine_plugin_model PluginUdl user_subject_udl user_id:integer  
subject_udl_id:integer
```

```
rake redmine:plugins:migrate
```

- **Crear Controlador del Plugin**

Es crearà un controlador per gestionar tot el plugin ja que no s'ha cregut convenient crear un controlador per a cada model, perquè les funcionalitats no són massa extenses. El controlador s'anomenarà PluginUdl.

```
ruby script/rails generate redmine_plugin_controller PluginUdl PluginUdl index  
newIndividualProject neuGroupProject
```

Es posen les tres accions que es necessitaran de base i a la vegada es crearan les vistes corresponents (Ja podem començar a programar)

### 4.6.3. Instal·lar PluginUdl

Aquests passos es faran després d'haver instal·lat i configurat el Redmine.

1. Copiar el directori del PluginUdl a la carpeta:

*/usr/local/share/redmine-2.3/plugins*

2. Com que el plugin requereix una migració, s'ha d'executar la següent comanda per crear les taules pertinents:

*rake redmine:plugins:migrate RAILS\_ENV=production*

3. Copiar tots els canvis realitzats al codi de Redmine (*Veure apartat 4.5.6*).
4. Inserir totes les dades de la UdL a les seves respectives taules (usuaris, assignatures, privilegis i la relació entre usuaris i assignatures).
5. Reiniciar Redmine.

### 4.6.4. Desinstal·lar PluginUdl

1. Executar la següent comanda per actualitzar la base de dades:

*rake redmine:plugins:migrate NAME=PluginUdl VERSION=0 RAILS\_ENV=production*

2. Eliminar el plugin de la carpeta plugins de Redmine:

*/usr/local/share/redmine-2.3/plugins*

3. Reiniciar el Redmine.

## 4.6.5. Modificacions Internes del codi de Redmine

Els avantatges de crear un Plugin els trobem en poder definir una eina adaptable a futures versions de Redmine, i perquè això passi també és molt important no modificar o alterar el codi intern.

Tot i així, s'ha modificat el model User de Redmine per poder definir les relacions entre les taules. Encara que la solució a estat modificar el model, també es pot incloure el codi al plugin perquè al fer la instal·lació s'insereixi automàticament.

Inserir el següent codi al Model User de Redmine:

```
- Ruta: /usr/local/share/redmine-2.3/app/models/user.rb
- Canvi:
  belongs_to :privilege_udl
  has_many :user_subject_udls
  has_many :subject_udls, :through => :user_subject_udls
```

## 4.6.6. Internacionalització

L'arxiu de configuració de la traducció del Redmine està dins de:

```
/usr/local/share/redmine-2.3/config/locales/.
```

La traducció dels plugins està al config del plugin:

```
/usr/local/share/redmine-2.3/plugins/plugin_udl/config/locales.
```

Per tal de internacionalitzar el plugin primer de tot s'haurà de mirar si existeixen els arxius de traducció dels llenguatges que volem determinar, en aquest cas només farem (Anglès, Català i Castellà). Per defecte només hi ha creat l'arxiu de traducció d'anglès que s'anomena: en.yml

S'han de crear els dos arxius que permetin la traducció al català i castellà:

```
ca.yml (català)
es.yml (castellà)
```

Sobretot els noms han de ser aquests ja que sinó quan es seleccioni una llengua no funcionarà la traducció.

Una vegada creats els documents, s'inclouran totes les traduccions del PluginUdl a cada

document, amb el seu llenguatge corresponent.

Exemple:

**- Arxiu ca.yml (català)**

ca:

text\_explicacio\_seleccio\_multiple: Per seleccionar més d'un professor utilitzar la tecla control o arrastra el ratoli

button\_save: Desa

button\_inserir\_projecte\_grup: Inserir Projecte/Grup

**- Arxiu en.yml (anglès)**

en:

text\_explicacio\_seleccio\_multiple: To select more than one teacher to use the control key or the mouse dragged

button\_save: Save

button\_inserir\_projecte\_grup: Insert Project/Group

**- Arxiu es.yml (espanyol)**

es:

text\_explicacio\_seleccio\_multiple: Para seleccionar mas de un profesor utilizar la tecla Control o Arrastrar el ratón

button\_save: Guardar

button\_inserir\_projecte\_grup: Insertar Proyecto/Grupo

**- Arxiu Vista**

<%= I(:text\_explicacio\_seleccio\_multiple) %>

<%= submit\_tag I(:button\_save) %>

<%= button\_to I(:button\_inserir\_projecte\_grup) %>

## 4.7. Estudi del procés d'autenticació d'un Usuari

Funcionament i connexió de com hauria de ser la autenticació de l'usuari:

1. L'alumne/professor s'autentica a l'aplicació.
2. L'aplicació comprova l'alumne/professor que entra, mitjançant LDAP (a la base de dades UdL).
3. Si l'usuari és la primera vegada que s'autentica a l'aplicació:

Si l'alumne/professor autenticat és vàlid (login i password estan bé), afegeix l'alumne a la Base de Dades de l'aplicació i el deixa entrar.

4. Si l'usuari ja ha entrat més d'un cop a l'aplicació:

El deixa entrar a l'aplicació.

5. El que es vol fer abans de que l'alumne/professor entri a l'aplicació és gestionar el seu espai de treball, afegint les assignatures, projectes,... relacionats amb cadascú. Això es faria mitjançant la base de dades que utilitza el Campus Virtual pel tema de donar d'alta assignatures. Hi ha dos opcions per a fer.

1. Agafar la informació de la mateixa base de dades del Campus, tot i que està carregada.
2. Bolcar la Base de Dades del Campus a un nova BD personalitzada per l'aplicació.

# CAPÍTOL 5. CONCLUSIONS I FUTURES LÍNIES DE TREBALL

## 5.1. Conclusions tecnològiques

En aquest TFC, Redmine ha demostrat ser una eina eficaç i força utilitzada en el món empresarial. El seu disseny entenedor, eficaç, de fàcil integració dels plugins,... ha permès poder crear un projecte ràpid amb les necessitats inicials sense haver de canviar especificacions durant el desenvolupament, ans el contrari, la seva versatilitat m'ha donat múltiples idees que podrien integrar-se en un futur.

La finalitat del projecte a estat crear un entorn real entre les pràctiques, els professors i els alumnes i crec que s'ha aconseguit. El plugin permet de forma molt ràpida generar les pràctiques com a projectes dins de Redmine. No crec que sigui un projecte lligat únicament al camp de la informàtica sinó que el podríem extrapolar a d'altres àmbits.

En la part d'usabilitat caldria fer un estudi més extens amb usuaris reals per determinar si realment el treball és una solució òptima per aplicar-lo a la universitat.

Tot i haver desenvolupat un plugin funcional pel Redmine crec que hi ha hagut coses que es podrien haver millorat, i tot seguit explico una mica les conclusions que he tret sobre el desenvolupament del PluginUdl.

El Gestor de Versions Subversion utilitzat per la configuració dels projectes de Redmine crec que en un futur s'hauria de canviar per diversos motius, i una de les alternatives podria ser el Git:

- Git és molt més ràpid que el Subversion.
- El rendiment del Git és molt més alt.
- Les característiques del projecte provoquen la creació d'un gran número de projectes i en conseqüència de Repositories. Els Repositories Git són molt més petits que el Subversion que requereix 30 vegades més d'espai per emmagatzemar el mateix historial.

Crec que el més important per un Gestor de Versions que treballi amb el Redmine que hem estudiat és el rendiment i l'espai que ocupen.

En la primera versió del PluginUdl vaig utilitzar el framework Prototype per desenvolupar Ajax a l'aplicació. En la última versió del PluginUdl he canviat Prototype per JQuery, ja que he trobat molta més informació i aquesta a partir de finals de 2011 passa a ser la llibreria oficial de Ruby on Rails. Crec que ha estat un gran canvi perquè he ampliat coneixements de JQuery, com l'ús de peticions Ajax, i a més s'ha facilitat la futura actualització del PluginUdl pel fet de que JQuery és una llibreria més universal.

En el desenvolupament de l'aplicació, un disseny més complert hauria ajudat al desenvolupament, ja que en casos puntuals s'ha hagut de redissenar alguna part i això ha suposat un augment del temps de dedicació en aquesta tasca.



El fet de treballar amb VMWare m'ha permès fer múltiples proves d'instal·lació de Redmine i del PluginUdl en diferents virtualitzacions. Això m'ha fet adonar de la gran dificultat que hi ha al instal·lar i configurar Redmine, cosa que no succeeix amb el plugin.

## 5.2. Conclusions Personals

Aquest treball m'ha servit com una aproximació a la gestió de projectes, des de la pròpia terminologia fins a les metodologies i estàndards propis, tant és així que en el món professional l'estic utilitzant per gestionar millores i errors amb els clients.

El fet de treballar amb un nou llenguatge i un nou framework com és Ruby on Rails m'ha servit per millorar, aprendre i saber adaptar-me a les noves tecnologies.

Crec que la implantació d'un Gestor de Projectes a la universitat és un aspecte molt important i a tenir en compte pel creixement dels estudiants, i no només pels futurs programadors sinó per tots els informàtics en general.

Pel que fa a l'opció d'instal·lar el PluginUdl crec que donaria moltes facilitats als professors i ajudaria a ordenar i classificar les pràctiques de les assignatures, dins del Redmine.

No totes les coses sobre Redmine són tan bones, en alguns moments la seva rigorositat fa, que el rendiment dels usuaris disminueixi, per això Redmine s'ha d'utilitzar amb moderació i no voler-lo utilitzar per tot. Un exemple seria voler fer un control estricte de les hores que ha passat un usuari amb una Tasca. És bo tenir un control de les hores però, no es pot permetre que l'usuari estigui més hores introduint-les que treballant-hi.

Per últim m'agradaria destacar una de les coses més importants que he après durant la realització d'aquest projecte, i és que quan vaig començar l'anàlisi, tenia clar que seria important conèixer un nou llenguatge com és Ruby i tot el que l'envolta, però ara, una vegada acabat el projecte puc dir que he après una de les bases imprescindibles per dur a terme un projecte: la gestió. Gestionar un projecte a tots els àmbits és fonamental perquè aquest es materialitzi de forma satisfactòria.

## 5.3. Futures línies de treball

Cal dir que la connexió automàtica entre la BD de Redmine i la UdL ara mateix no existeix i per tant una futura ampliació seria connectar la base de dades de la UdL i Redmine, similar a Sakai. Seria bo crear una eina que traspasés diàriament tota la informació necessària al Redmine (usuaris, assignatures, la seva relació i els seus privilegis).

També una de les coses importants que caldria fer seria poder definir les característiques dels projectes dins del PluginUdl, i així controlar en cada pràctica els mòduls més adients (Wiki, Repository, Calendar, News, Documents,...).

S'hauria, a més, de dissenyar millor el theme Udl creat com a disseny del Redmine, ja que visualment té algunes coses a millorar.

Una de les coses que havia pensat mentre desenvolupava l'aplicació seria també poder enviar correus als alumnes cada cop que es creés un projecte.

També seria bo validar l'identificador del projecte. Aquest últim més que una futura línia de treball és un bug que m'he trobat durant la fase de proves del PluginUdl. L'identificador es basa amb el codi d'assignatura i és molt difícil que es dupliqui, però podria passar que un professor per equivocació crees dos pràctiques 1 en una mateixa assignatura i aleshores l'identificador es duplicaria i no es crearien els projectes.

Malgrat s'hagin introduït diferents camps per definir si una assignatura està deshabilitada o finalitzada (`academic_year`, `enabled`) no s'han utilitzat en el desenvolupament. En futures ampliacions s'hauria de filtrar les assignatures per aquests camps.

Finalment es podria integrar l'eina Redmine per a Mòbils.



# Bibliografia

La font principal de la bibliografia en aquest projecte a estat la Web de Redmine (<http://www.redmine.org/>). En ella hi podem trobar la documentació referent a la configuració de Redmine. Tot seguit s'indiquen alguns dels links més importants que s'han utilitzat:

Per crear els Repositoris automàticament:

[http://www.redmine.org/projects/redmine/wiki/HowTo\\_Automate\\_Repositories\\_creation](http://www.redmine.org/projects/redmine/wiki/HowTo_Automate_Repositories_creation)

Per autenticar els usuaris al svn:

[http://www.redmine.org/projects/redmine/wiki/HowTo\\_configure\\_Redmine\\_for\\_advanced\\_Subversion\\_integration](http://www.redmine.org/projects/redmine/wiki/HowTo_configure_Redmine_for_advanced_Subversion_integration)

Per crear el Plugin:

[http://www.redmine.org/projects/redmine/wiki/Plugin\\_Tutorial](http://www.redmine.org/projects/redmine/wiki/Plugin_Tutorial)

Opció d'autenticació LDAP:

[http://www.redmine.org/wiki/redmine/Repositorieses\\_access\\_control\\_with\\_apache\\_mod\\_dav\\_svn\\_and\\_mod\\_perl#optional-LDAP-Authentication](http://www.redmine.org/wiki/redmine/Repositorieses_access_control_with_apache_mod_dav_svn_and_mod_perl#optional-LDAP-Authentication)

Per la configuració del correu electrònic:

<http://www.redmine.org/projects/redmine/wiki/EmailConfiguration>

Altres webs d'interès que també s'han utilitzat per desenvolupar el projecte són:

Jquery i Ajax:

<http://www.desarrolloweb.com/articulos/uso-ajax-jquery.html>

Funcionament de RubyGems:

<https://rubygems.org/>

RubyMine:

<http://blog.jetbrains.com/ruby/>

Finalment també s'han utilitzat els llibres següents:

David A. Black. *Ruby Techniques for Rails Developers. Ruby for Rails*. Manning Publications Co. Greenwich.

Noel Rappin. *Professional Ruby on Rails™*. Wiley Publishing, Inc.